K2 Learning

Leave Request (Basic)

May 2019

Leave Request (Extended Version)

In this intermediate-level tutorial, you will build a more advanced Leave Request approval application by extending the Leave Request (Basic Version) application. You will learn more advanced concepts in working with SmartObjects such as how to integrate with external systems by using SmartObjects, how to add more advanced workflow functions like reminders and rework loops, and add more advanced functionality to the forms used in the application, for example retrieving data from an external system.

If you want to learn a little more about the application you are about to build and how it will behave, review the design of the application in the topic Leave Request (Extended Version) Application Design.

> **Note**
> The Leave Request (Extended Version) tutorial builds upon the basic version tutorial. You must have the completed components from the basic version in place, before beginning this tutorial. To build the basic version, please see the Leave Request (Basic Version) tutorial

Steps

1. Add the Approver Comments Property to the Leave Request SmartObject
2. Create a SQL Server Service Instance
3. Create a new SmartObject from a Service Instance
4. Add the Approver Comments Control to the Leave Request Item View and Bind the Leave Type Drop-Down List
5. Add and Configure Email Steps for the Manager Outcomes
6. Add the Rework Outcome and Requester Rework Task
7. Add a Reminder to the Manager Approval Step
8. Deploy the Leave Request Workflow
9. Edit a Rule to Enable the Approver Comments for the Workflow Task State
10. Add a Save Method to Update the SmartObject with Manager Changes
11. Add a Clear Method to Clear Form Fields After Submit
12. Test the Leave Request (Extended Version) Application

(Optional) Clean your K2 environment by deleting application artifacts

First Step: 1. Add the Approver Comments Property to the Leave Request SmartObject

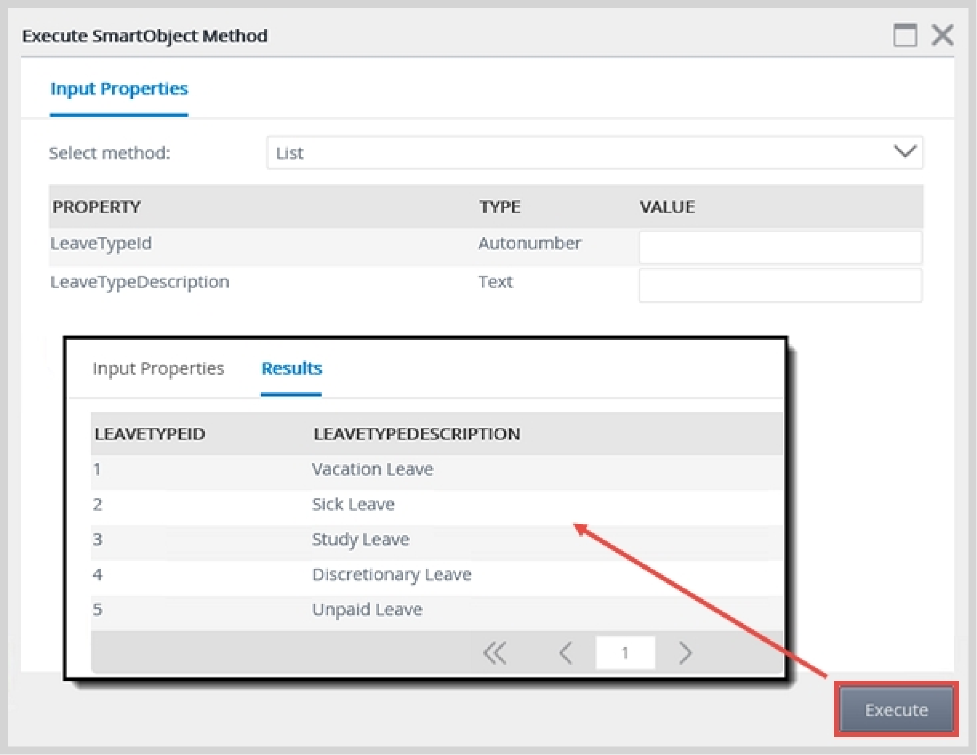
Leave Request (Extended Version): Application Design

The Leave Request (Extended Version) application begins with editing the existing Leave Request SmartObject, then creating a new SmartObject that leverages data from an external data source. This data will replace the static values in the Leave Type drop-down list. You will edit the Leave Request Workflow and add a rework loop. This loop sends the request back to originator if the approving manager has questions or needs more information. On the Leave Request Form, you will work with rules and states, where you will further customize the user interfaces and for behavior.

Data

To extend the **data** component, you will edit the Leave Request SmartObject and add a new property. The property, *Approver Comments*, will allow the approving manager to add any questions or comments they may have. This step will show you how to edit an existing SmartObject. Next, you will add a new **service instance**, or connection, to an external data source. The data source, an Azure SQL Server database, contains a table of leave types, which will come into play for extending the forms component. You will generate a new **Leave Types SmartObject** from the service instance.

The results that are returned from the external data source



Execute SmartObject Method

Input Properties

Select method: List

PROPERTY	TYPE	VALUE
LeaveTypeId	Autonumber	
LeaveTypeDescription	Text	

Input Properties | Results

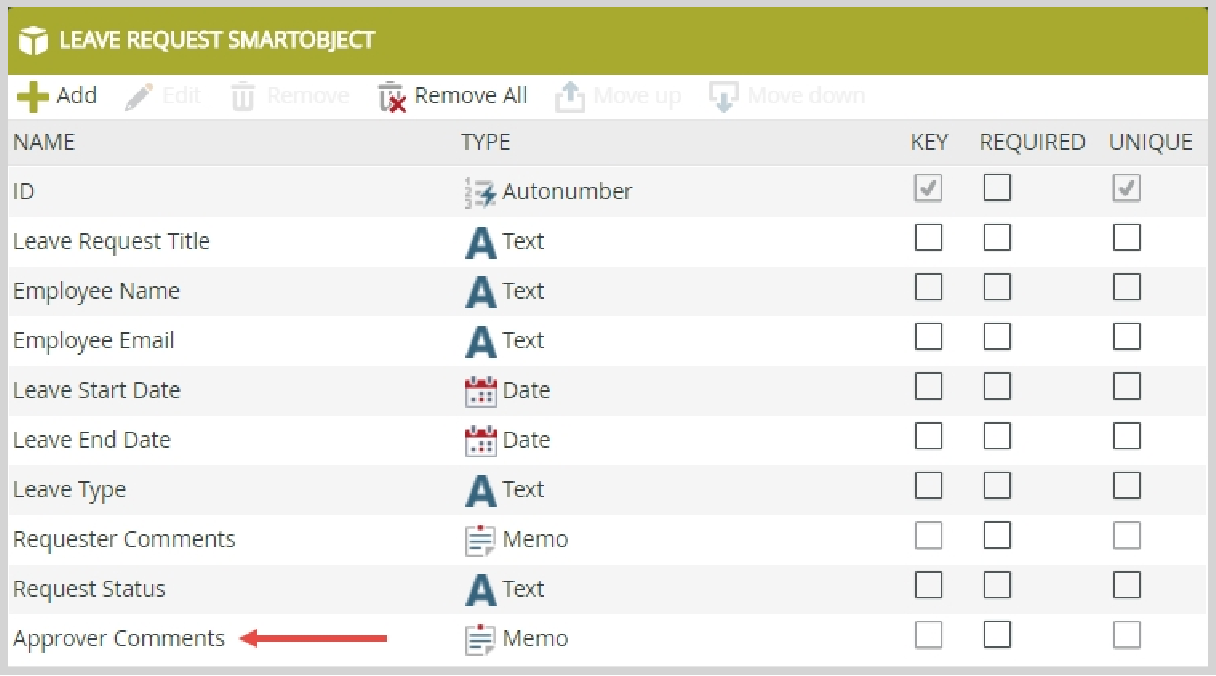
LEAVETYPEID	LEAVETYPEDESCRIPTION
1	Vacation Leave
2	Sick Leave
3	Study Leave
4	Discretionary Leave
5	Unpaid Leave

Leave Request SmartObject

This tutorial will expand your knowledge of SmartObjects and how you can use SmartObjects to interact with other systems. SmartObjects are the connectors or the "middle layer" between providers of data and consumers of data. Providers of data are where the data lives. Examples include SharePoint lists, SQL Server tables, and Active Directory. Consumers of data are the objects that use data. Examples include form fields and workflows.

The first task is to add a new property to the **Leave Request SmartObject**. The new property, *Approver Comments*, is a memo data type that will capture any comments or questions the approving manager has.

The Leave Request SmartObject with the new Approver Comments property



LEAVE REQUEST SMARTOBJECT

NAME	TYPE	KEY	REQUIRED	UNIQUE
ID	Autonumber	☑	□	☑
Leave Request Title	Text	□	□	□
Employee Name	Text	□	□	□
Employee Email	Text	□	□	□
Leave Start Date	Date	□	□	□
Leave End Date	Date	□	□	□
Leave Type	Text	□	□	□
Requester Comments	Memo	□	□	□
Request Status	Text	□	□	□
Approver Comments	Memo	□	□	□

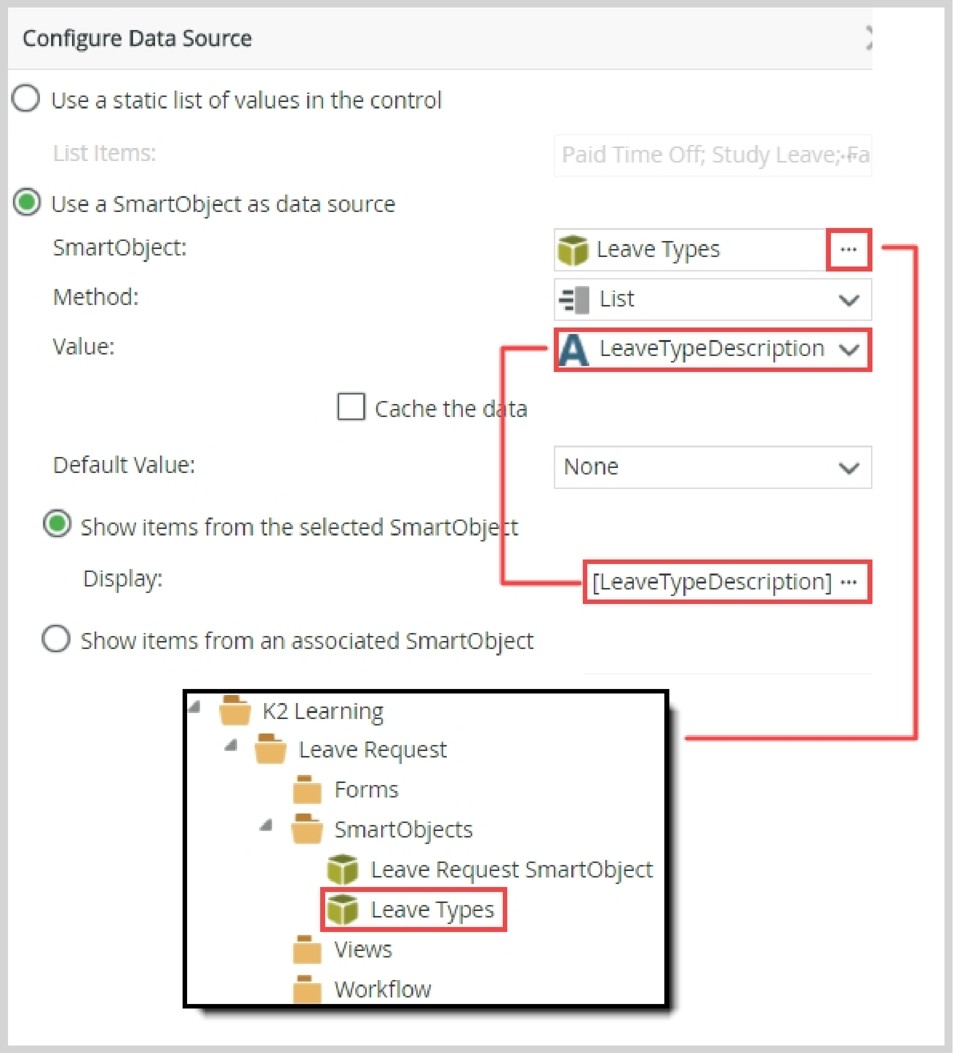
The second task is to create a new SmartObject, along with the underlying configuration, to connect to an external SQL database. First, you will create a service instance of a SQL Server service type. You will configure the service instance for an Azure SQL database. Finally, you will generate a new SmartObject (**Leave Types SmartObject**) from the service instance, that will return a list of leave types. In the extended user interfaces tutorial, you will replace the static values for the Leave Type drop-down list with the values returned from the Leave Types SmartObject.

> **Note**
> In a K2 environment, registering a service instance for a particular system only needs to be performed once. If you are using a shared environment (for example, if you are completing this exercise in your own K2 environment), this step only needs to be performed once and may already have been completed by another user. If you prefer to create your own service instances, use a unique identifier for the service instance display names and system names. Otherwise, use the service instance that already exists.

Forms

To extend the **forms** component, you will bind the Leave Type drop-down list to the new Leave Types SmartObject. In the basic version, the drop-down list contains static values. Now, the drop-down list will contain the values from the new SmartObject. There is an advantage to using SmartObject data for form controls. If you need to edit the leave type line items, you can edit the SmartObject itself and any controls that use that data will be immediately updated. This also keeps the leave type values consistent where used.

Binding the Leave Type control to the Leave Types SmartObject



Configure Data Source

○ Use a static list of values in the control

List Items: Paid Time Off; Study Leave;-Fa

● Use a SmartObject as data source

SmartObject: Leave Types

Method: List

Value: LeaveTypeDescription

□ Cache the data

Default Value: None

● Show items from the selected SmartObject

Display: [LeaveTypeDescription]

○ Show items from an associated SmartObject

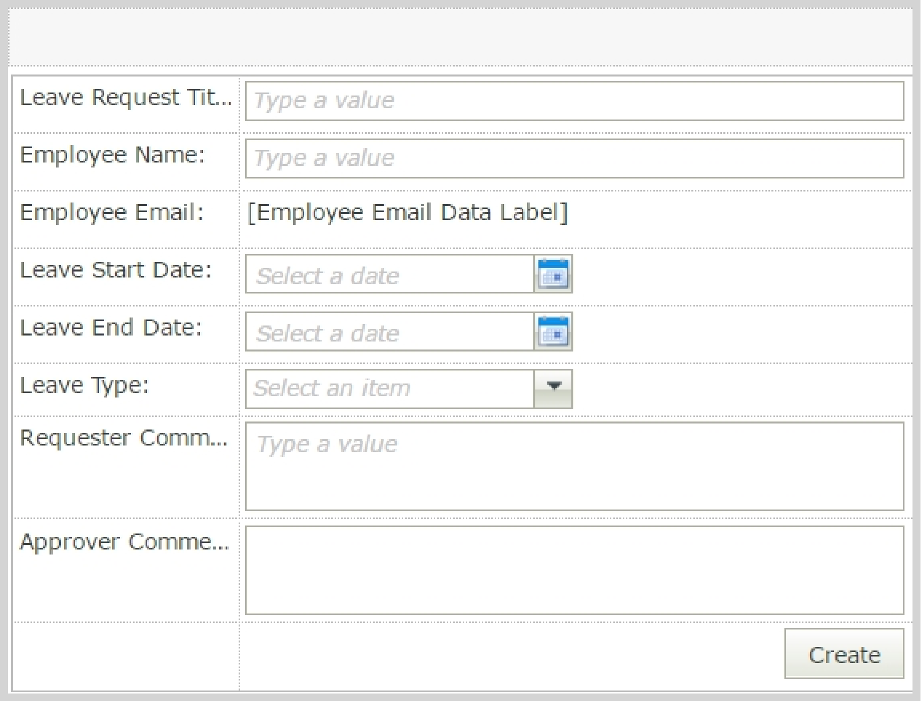
Leave Request Item View

Edit the **Leave Request Item View**. You will bind the **Leave Type** drop-down list control to the new **Leave Types SmartObject**. The drop-down list contains static values for the list items. The Leave Types SmartObject returns a list of leave types found in the Azure SQL database, which will replace the static values. This will create dynamic values for the drop-down list.

Next, you will add the new **Approver Comments** property to the Leave Request Item View, then make the property read-only for the requester. In a later step, you will enable the property so that the approving manager can add any questions or comments they have.

Finally, you will delete the **Request Status** label and control. The workflow updates the status value as it moves along the outcome lines. Since it may be confusing to have the field displayed on the view, you will remove it.

The extended Leave Request Item View



Leave Request Tit...	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comm...	Type a value
Approver Comme...	
	Create

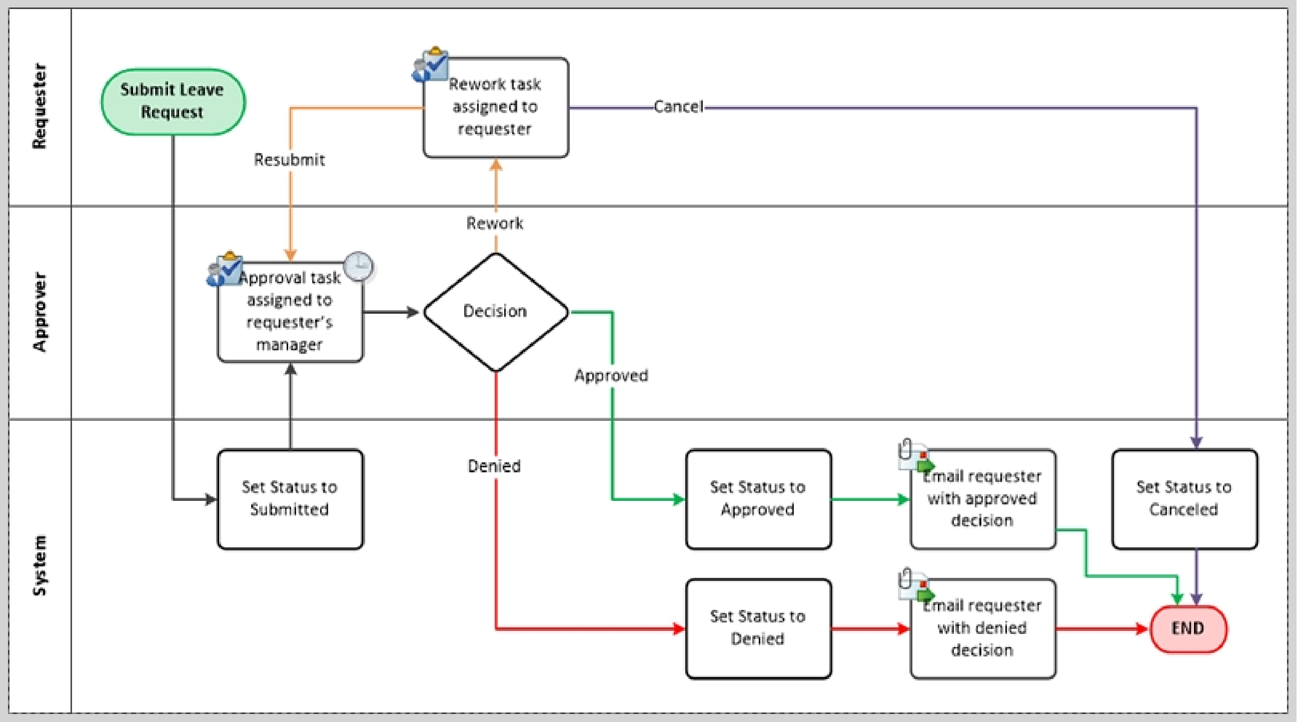
Workflow

To extend the **workflow** component, you will add a third action to the Manager Approval task. This action allows the manager to send the request back to the requester for rework. The requester can either resubmit the request or cancel the request. The workflow is further expanded on below. You will also add Send Email steps for the Approved and Denied outcomes.

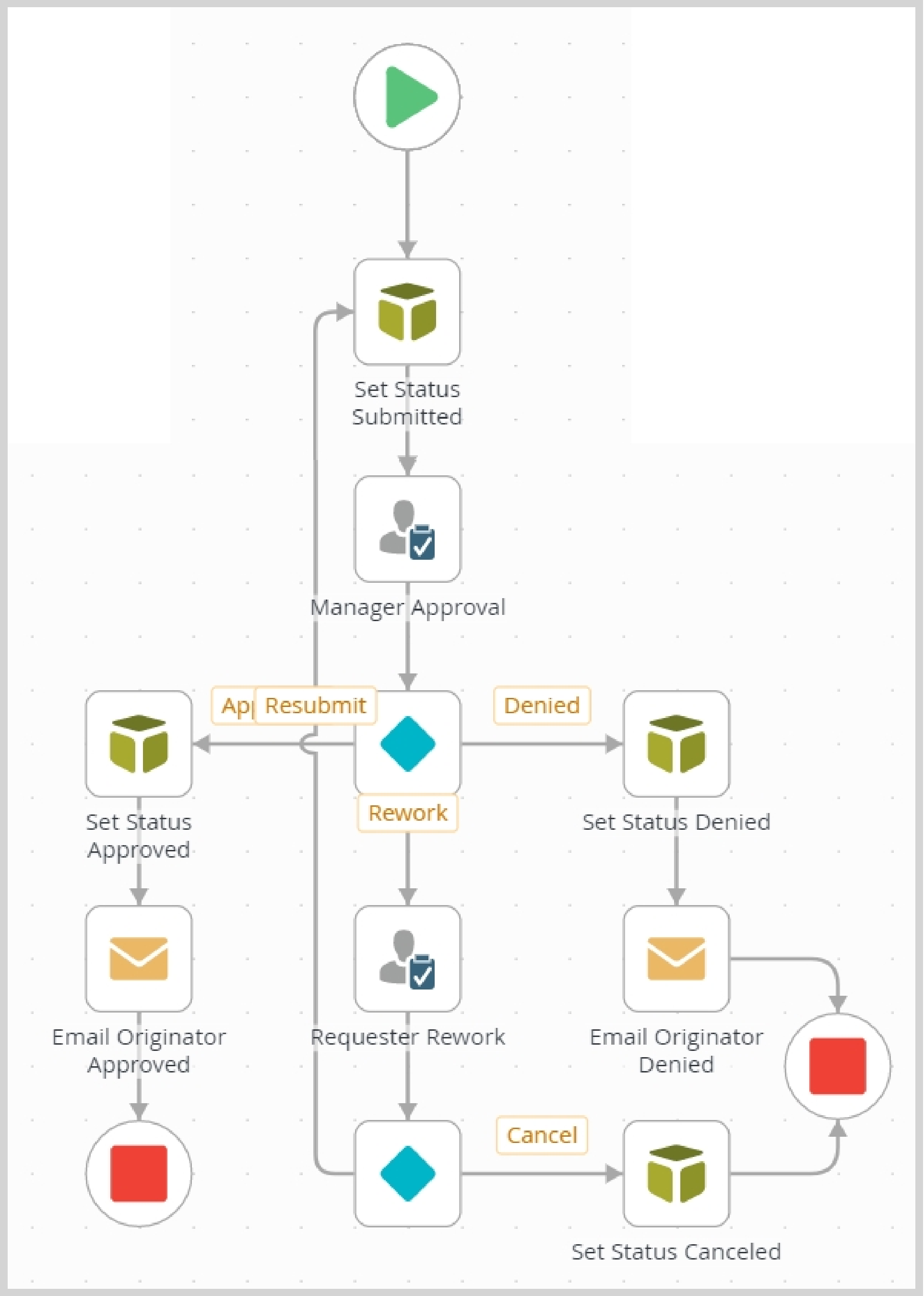
Leave Request Workflow Design

As you learned in the basic version tutorial, mapping out your workflow is key to organizing the components you need for the application. For the expanded application, you will add email, as well as a new outcome, rework. Once again, the flowchart is in a swim-lane format with the first lane representing the requester's actions and tasks. The second lane represents the approving manager's actions and tasks, while the third lane represents system tasks performed by the K2 server.

The Leave Request Workflow represented as a flowchart.



The extended Leave Request Workflow

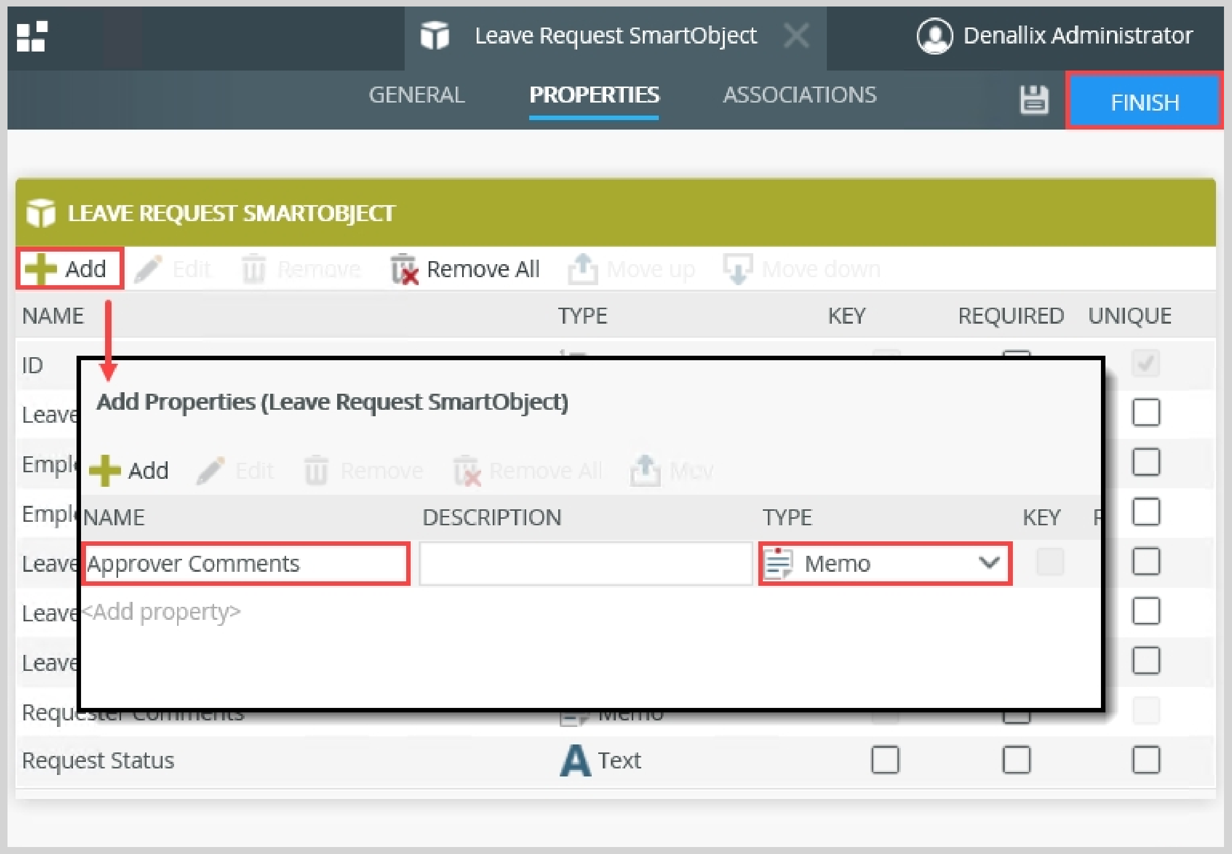


First Step: 1. Add the Approver Comments Property to the Leave Request SmartObject

1. Add the Approver Comments Property to the Leave Request SmartObject

In this step, you will add a new property (Approver Comments) to the existing Leave Request SmartObject. This allows the approving manager to enter any questions or comments they have about the request. This step demonstrates how to edit an existing SmartBox SmartObject. Adding new properties to an existing SmartBox SmartObject is relatively simple. You should take caution however, when deleting properties. Any controls or rules that rely on those deleted properties will likely cause errors!

1. In K2 Designer, edit the **Leave Request SmartObject** and add a **Memo** type property called *Approver Comments*.

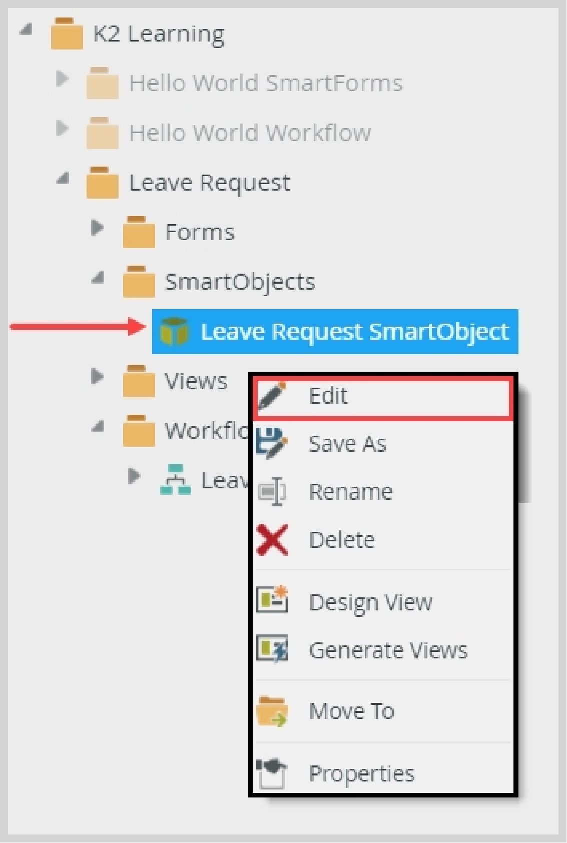


NAME	DESCRIPTION	TYPE	KEY
Approver Comments		Memo	□

a. Launch **K2 Designer**

> **Note**
> If you are unsure of how to launch the **K2 Designer** site, see Accessing K2 Sites.

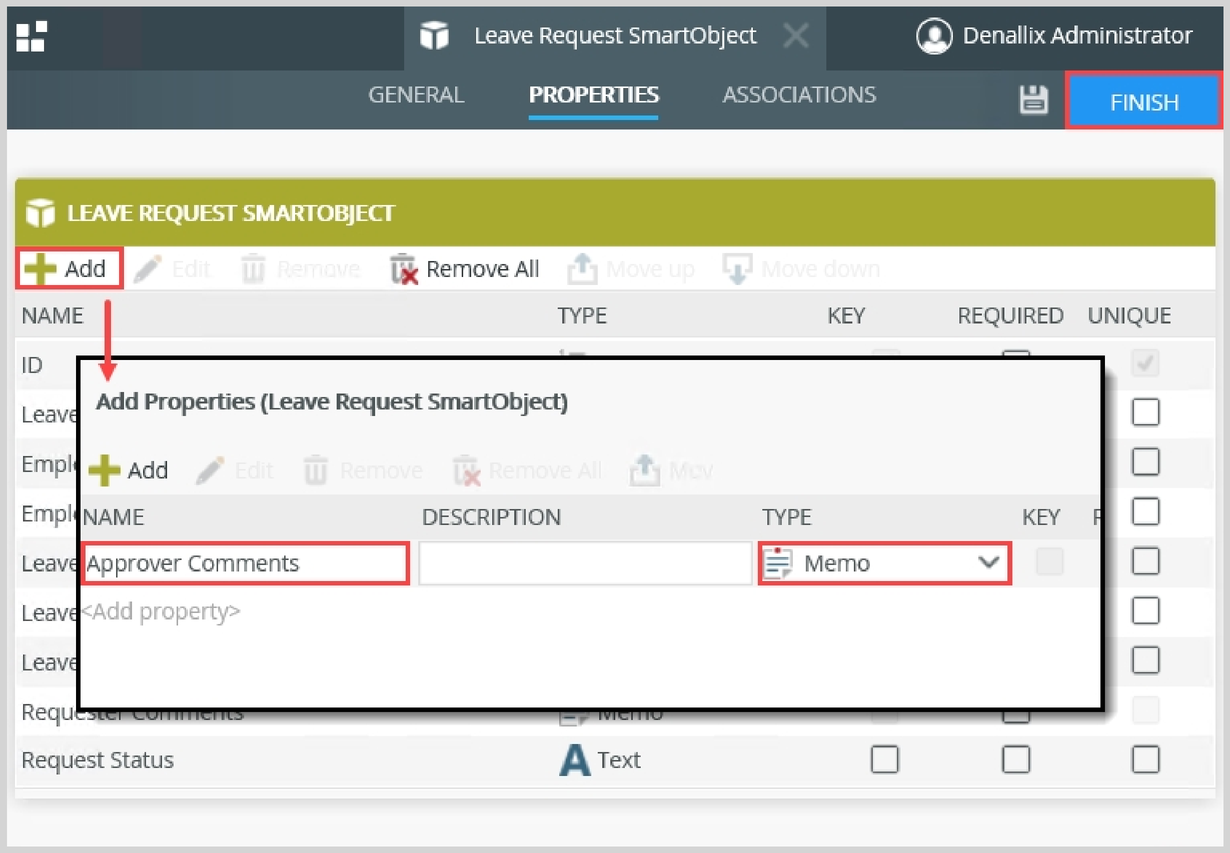
b. Browse to the category explorer, and then expand the **All Items** > **K2 Learning** > **Leave Request** > **SmartObjects** categories. Right-click the **Leave Request SmartObject** and select **Edit**.



c. Next, you will add the memo property to the SmartObject. (In a later step, you will edit the Leave Request Item View and manually add this new property to the view). On the PROPERTIES screen, click **Add**. For the **Name**, enter

Approver Comments

then select the **Memo** option from the data **Type** drop-down. Click **OK** (at the bottom of the Add Properties screen), then click **FINISH** to save the new property. Remember, when working with SmartObjects in the K2 Designer, finishing the SmartObject "publishes" it to the K2 server. The new property is now available for use in views, forms and workflows.



Add Properties (Leave Request SmartObject)

NAME	DESCRIPTION	TYPE	KEY
Approver Comments		Memo	□

Review

In this step, you learned how to edit an existing SmartBox SmartObject and add a new property. Clicking the Finish button publishes the SmartObject to the K2 server. Be careful when deleting properties from a SmartObject. Workflows and other components bound to deleted properties will produce errors! In the next step, you will create a new service instance to an external data source.

Next Step: 2. Create a SQL Server Service Instance

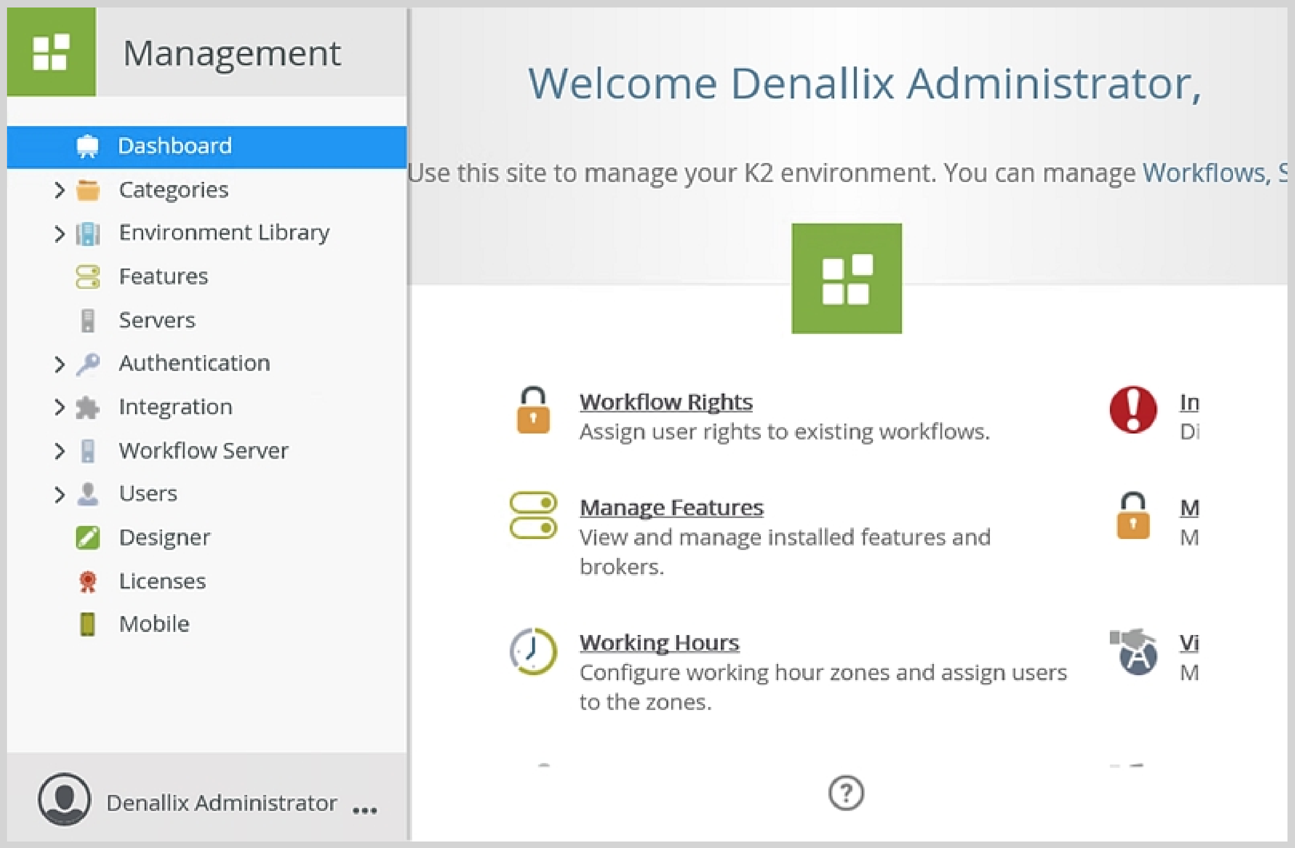
2. Create a SQL Server Service Instance

In this step, you create a new service instance of the SQL Server Service type. The service instance is essentially a *connection* configuration that targets a database on Azure SQL. The database provides a list of leave types that you will use in your Leave Request Form.

1. Launch the **K2 Management** site.
    a. Launch the **K2 Management** site.

    > **Note**
    > If you are unsure of how to launch the **K2 Management** site, see Accessing K2 Sites.

    b. The management site opens with the dashboard.



Welcome Denallix Administrator,

Use this site to manage your K2 environment. You can manage Workflows, S

Workflow Rights
Assign user rights to existing workflows.

Manage Features
View and manage installed features and brokers.

Working Hours
Configure working hour zones and assign users to the zones.

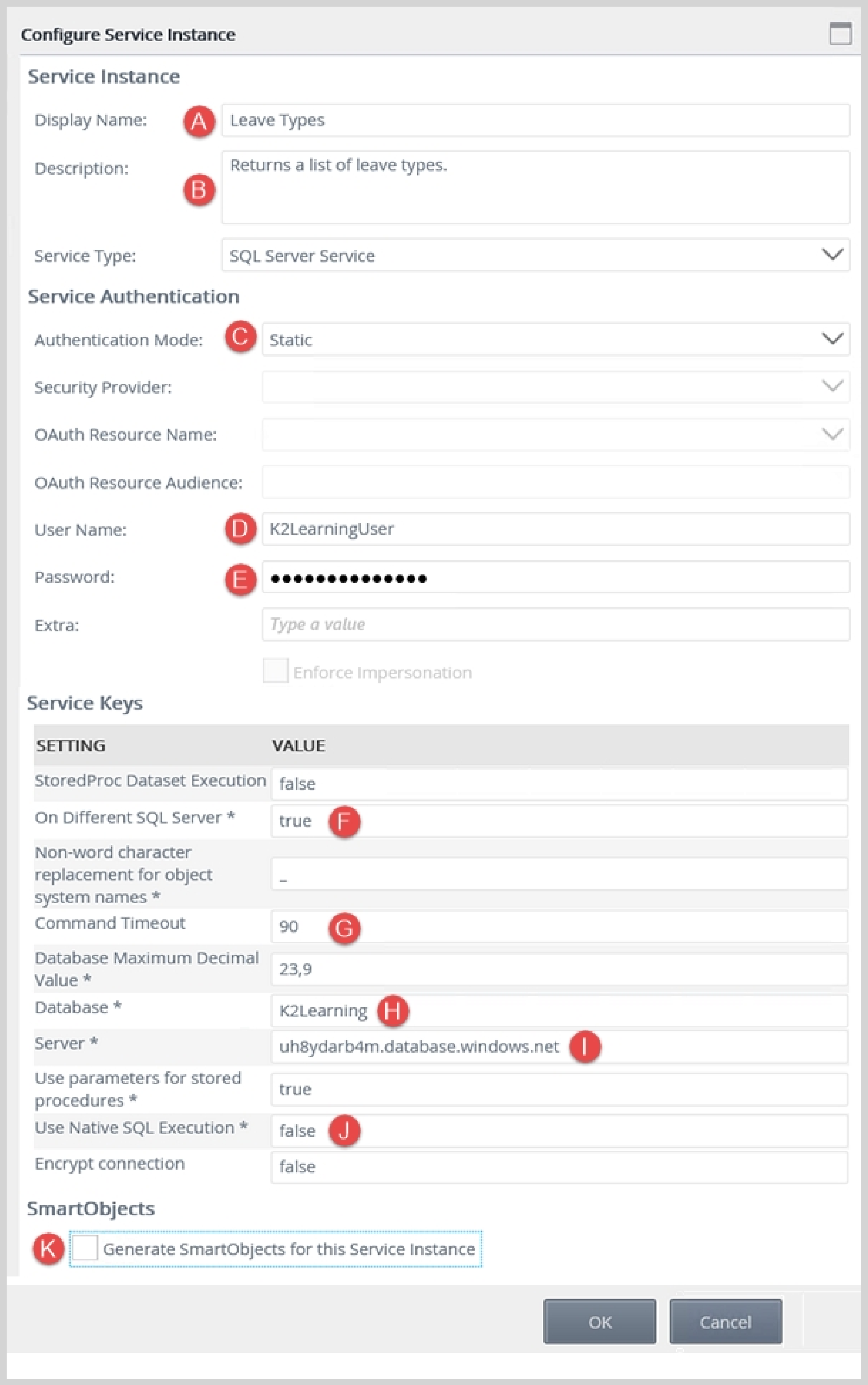
2. Add a new service instance of the **SQL Server Service** type using the table below as a guide for the properties. If a property is not shown, you can assume the default value.

    > **Tip**
    > In a shared environment, you only need to create the service instance once. If the service instance already exists and you want to create your own, use a unique identifier of some kind. One example might be to add your initials to the service instance name.

    > **Note**
    > The database you will be connecting to is hosted on the internet as a SQL Azure database. This database should be accessible through port 1433, and unless your organization has strict firewall policies in place, it should be available for use anywhere. If you need to host this SQL database internally, you can download a SQL script to recreate the database from the following location: http://help.k2.com/files/8553. Contact a

> member of your application development team for help, if necessary. You will need to adjust the Service Instance's **Database** and **Server** keys and potentially the **authentication method** to connect to a local version of this SQL Database.

	Field Name	Setting
A	Display Name	*Leave Types*
B	Description	*Returns a list of leave types.*
	Service Type	SQL Server Service (default)
C	Authentication Mode	Static
D	User Name	*K2LearningUser*
E	Password	*K2LearningPass*
F	On Different SQL Server	*true*
G	Command Timeout	*90*
H	Database	*K2Learning*
I	Server	*uh8ydarb4m.database.windows.net*
J	Use Native SQL Execution	*false*
K	Generate SmartObjects for this Service Instance	UNCHECKED



Configure Service Instance

Service Instance

Display Name:	Leave Types
Description:	Returns a list of leave types.
Service Type:	SQL Server Service

Service Authentication

Authentication Mode:	Static
Security Provider:	
OAuth Resource Name:	
OAuth Resource Audience:	
User Name:	K2LearningUser
Password:	•••••••••••••••
Extra:	Type a value

□ Enforce Impersonation

Service Keys

SETTING	VALUE
StoredProc Dataset Execution	false
On Different SQL Server *	true
Non-word character replacement for object system names *	_
Command Timeout	90
Database Maximum Decimal Value *	23,9
Database *	K2Learning
Server *	uh8ydarb4m.database.windows.net
Use parameters for stored procedures *	true
Use Native SQL Execution *	false
Encrypt connection	false

SmartObjects

□ Generate SmartObjects for this Service Instance

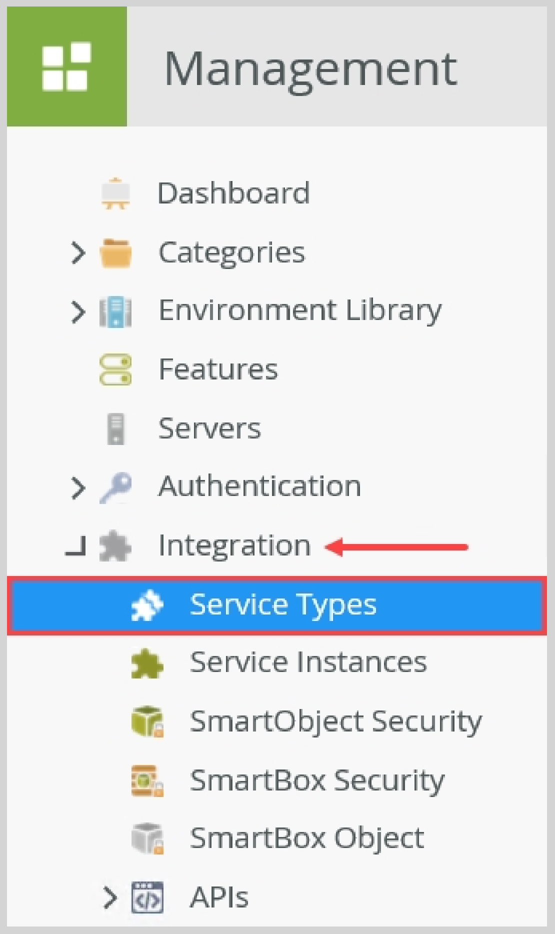
a. Next, you will create a new instance of the SQL Server Service type. To summarize this process, you are creating a connection to a specific data source (Azure SQL Database) so that you can retrieve the properties and methods (Leave Types) from that data source to use in your application (Leave Type drop-down list). K2 has a large number of out-of-the-box service types that you can use to connect to other data sources and you can create your own as well.
Expand the **Integration** category, then click **Service Types**.

Note

Service Types, Brokers, Instances, and SmartObjects

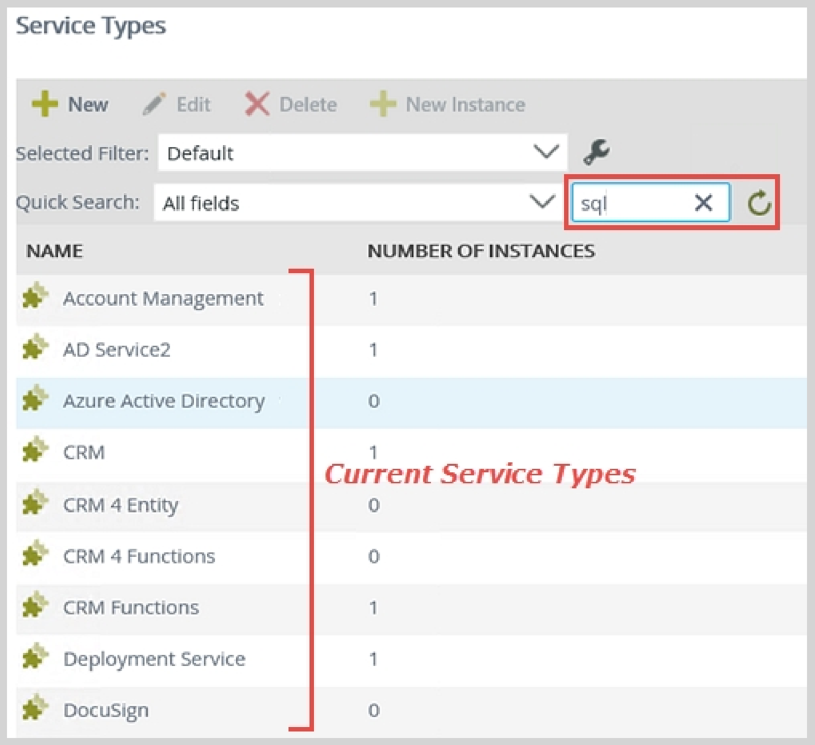
Service types, service brokers, and service instances are the elements that allow K2 to interact with other systems, and which form the base layer for SmartObjects. The following is a brief overview of each concept:

- **Service Type**: A service type is a pointer to a broker file for a specific system or data source. Examples include: SQL Server, SharePoint, CRM, and web services. Each service type has an underlying service broker associated with it.
    - **Service Broker**: A file that contains the logic needed to interact with a specific system. Each service type has its own requirements for interacting with the system. For example, what type of authorization will the system allow? What type of data is contained in the system?
- **Service Instance**: A service instance is a single connection to a data source, and is based on the service type. The service instance uses the requirements defined by the broker to connect to the target data source. For example, you might have an instance of a SQL Server service type. The instance is specific to a single SQL database. If you have multiple databases, you need multiple instances. From the instances, you can then generate SmartObjects.
- **SmartObjects**: The middle layer that allows interaction between a K2 object (form, view, workflow) and the target data source. For example, you have a form bound to a SQL SmartObject. When you submit the form, the SmartObject creates a new record or it updates an existing record in the SQL table. The SmartObject connects K2 to the system via the service layer.



b. The available service types appear in the **Service Types** central pane. K2 provides many of these service types out-of-the-box, however, there may be additional service types added along the way. You will be adding a service instance of the SQL Server Service type for this step. To help you locate the correct service type, enter
sql

into the search text box, then click the green **refresh** icon.



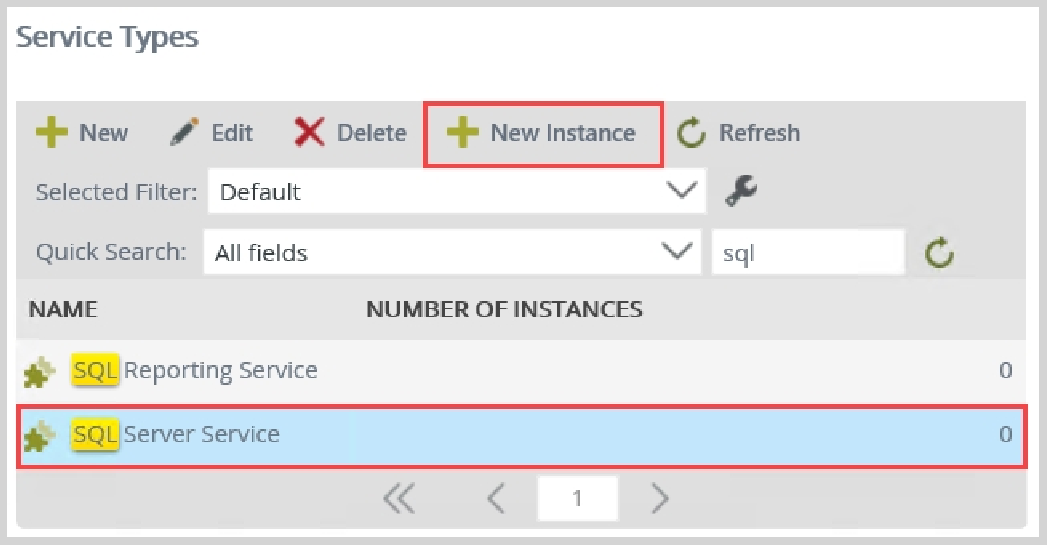
Service Types

Selected Filter: Default

Quick Search: All fields sql

NAME	NUMBER OF INSTANCES
Account Management	1
AD Service2	1
Azure Active Directory	0
CRM	1
CRM 4 Entity	0
CRM 4 Functions	0
CRM Functions	1
Deployment Service	1
DocuSign	0

c. K2 returns any service types with "SQL" in the name. (Depending on your environment, you may see additional service types.) Select **SQL Server Service**, then click the **New Instance** button. In this step, you are creating a new instance of the SQL Server Service type.



Service Types

Selected Filter: Default

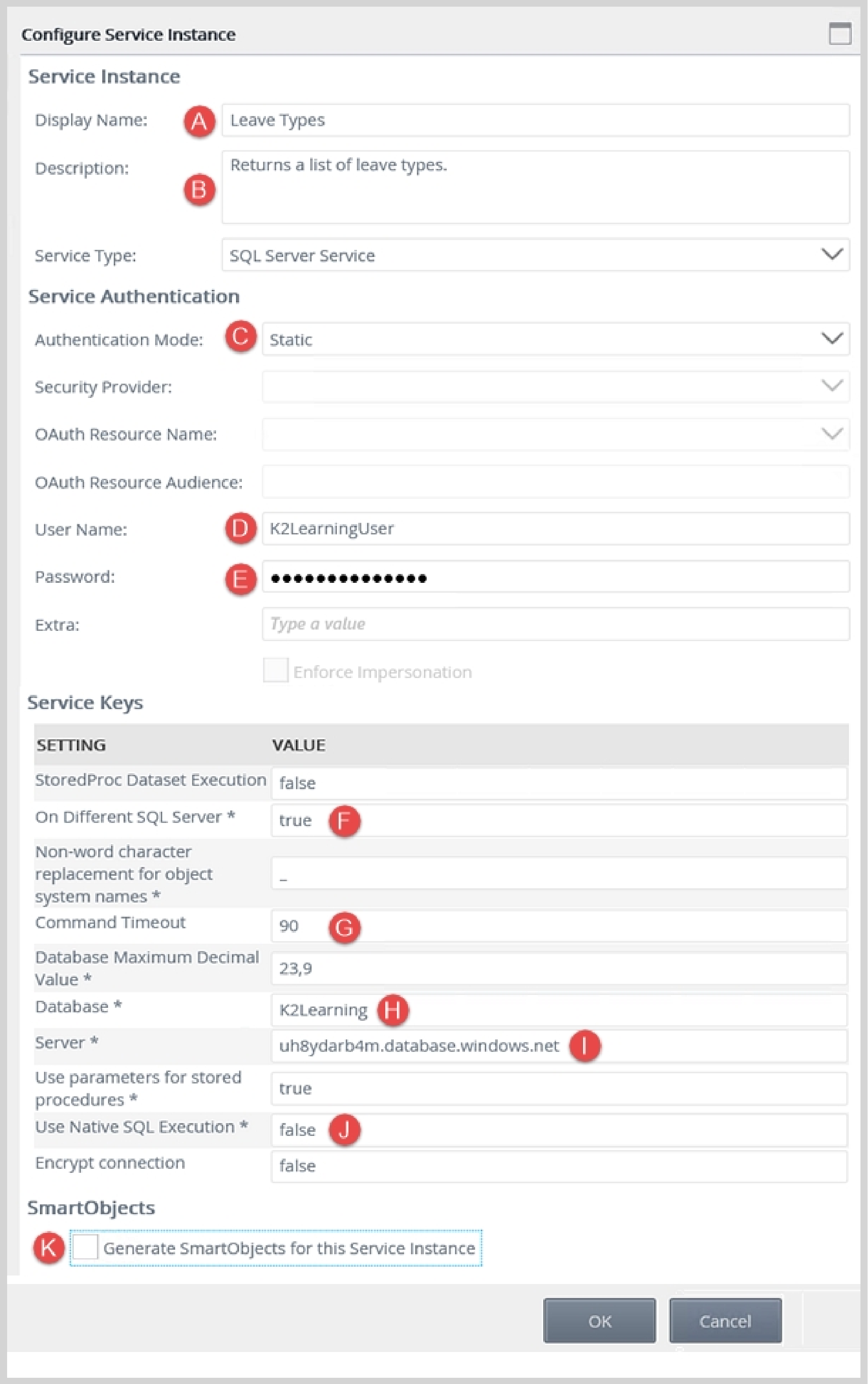
Quick Search: All fields sql

NAME	NUMBER OF INSTANCES
SQL Reporting Service	0
SQL Server Service	0

d. Configure the service instance using the table below for reference. There is also a reference image below the table. If a setting is not specified in the table, then assume the default value. Click **OK** after you have set the fields. In this step, you are adding the configuration information necessary to connect to the external Azure SQL database.

	Field Name	Setting

A	Display Name	*Leave Types*
B	Description	*Returns a list of leave types.*
	Service Type	SQL Server Service (default)
C	Authentication Mode	Static
D	User Name	*K2LearningUser*
E	Password	*K2LearningPass*
F	On Different SQL Server	*true*
G	Command Timeout	*90*
H	Database	*K2Learning*
I	Server	*uh8ydarb4m.database.windows.net*
J	Use Native SQL Execution	*false*
K	Generate SmartObjects for this Service Instance	UNCHECKED



Configure Service Instance

Service Instance

Display Name: Leave Types

Description: Returns a list of leave types.

Service Type: SQL Server Service

Service Authentication

Authentication Mode: Static

Security Provider:

OAuth Resource Name:

OAuth Resource Audience:

User Name: K2LearningUser

Password: ••••••••••••••

Extra: *Type a value*

□ Enforce Impersonation

Service Keys

SETTING	VALUE
StoredProc Dataset Execution	false
On Different SQL Server *	true
Non-word character replacement for object system names *	_
Command Timeout	90
Database Maximum Decimal Value *	23,9
Database *	K2Learning
Server *	uh8ydarb4m.database.windows.net
Use parameters for stored procedures *	true
Use Native SQL Execution *	false
Encrypt connection	false

SmartObjects

□ Generate SmartObjects for this Service Instance

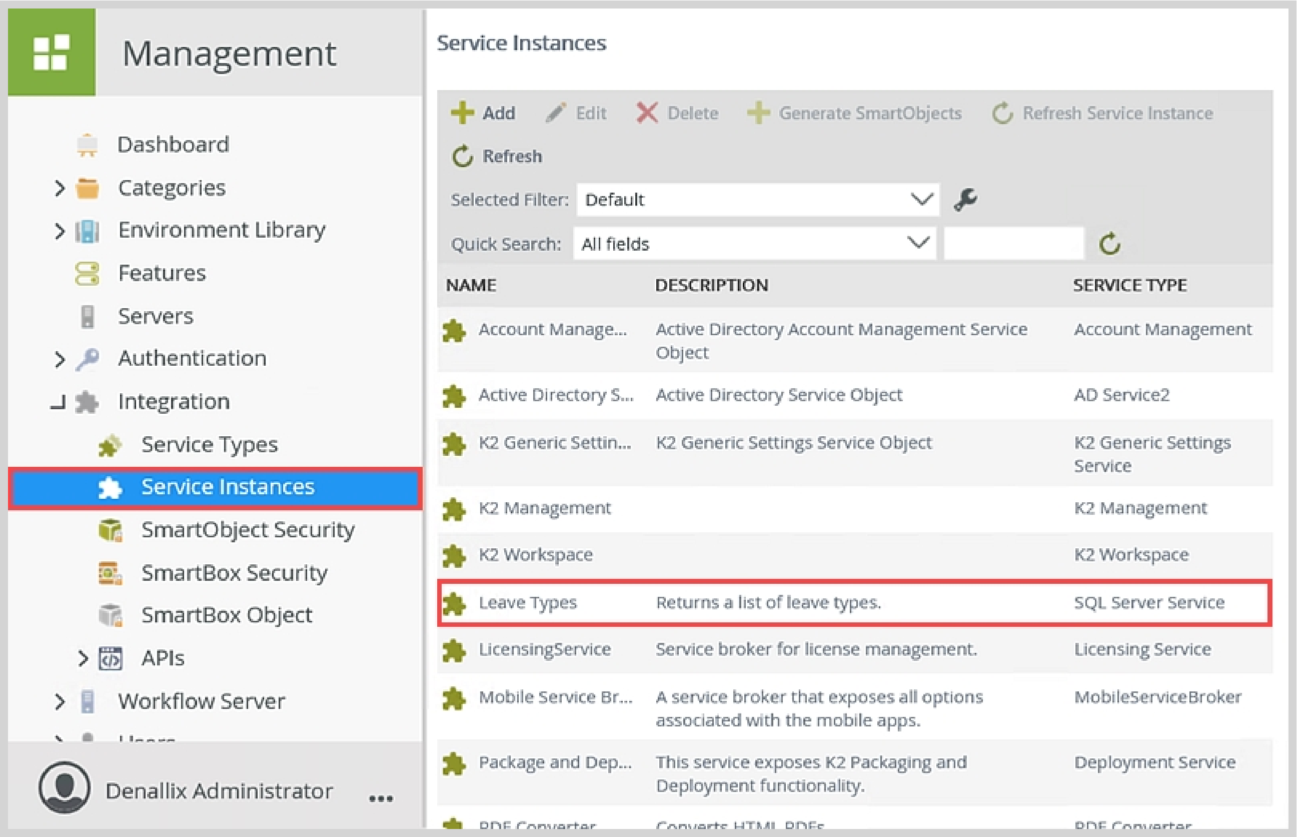
Note

Notice that you did not check the box to **Generate SmartObjects for this Service Instance**. In the next topic, you will create a new SmartObject manually from this service instance. You will create the SmartObject, then add the methods and properties for retrieving leave types. By checking this box, K2 creates SmartObjects for you, based on the tables and views it discovers in the database. Checking this box has the potential to create many SmartObjects from a

> single service instance. Creating the SmartObject manually allows you to create one SmartObject with the properties and methods you need.

e. You should see a confirmation dialog box. Click **OK**.

f. Take a moment to locate the new service instance you created. Still in the **Integration** node, click **Service Instances**. Scroll down and locate the **Leave Types** service instance. Notice the name, description, and service type correspond to your configuration settings.



NAME	DESCRIPTION	SERVICE TYPE
Account Manage...	Active Directory Account Management Service Object	Account Management
Active Directory S...	Active Directory Service Object	AD Service2
K2 Generic Settin...	K2 Generic Settings Service Object	K2 Generic Settings Service
K2 Management		K2 Management
K2 Workspace		K2 Workspace
Leave Types	Returns a list of leave types.	SQL Server Service
LicensingService	Service broker for license management.	Licensing Service
Mobile Service Br...	A service broker that exposes all options associated with the mobile apps.	MobileServiceBroker
Package and Dep...	This service exposes K2 Packaging and Deployment functionality.	Deployment Service
PDF Converter	Converts HTML PDFs	PDF Converter

Review

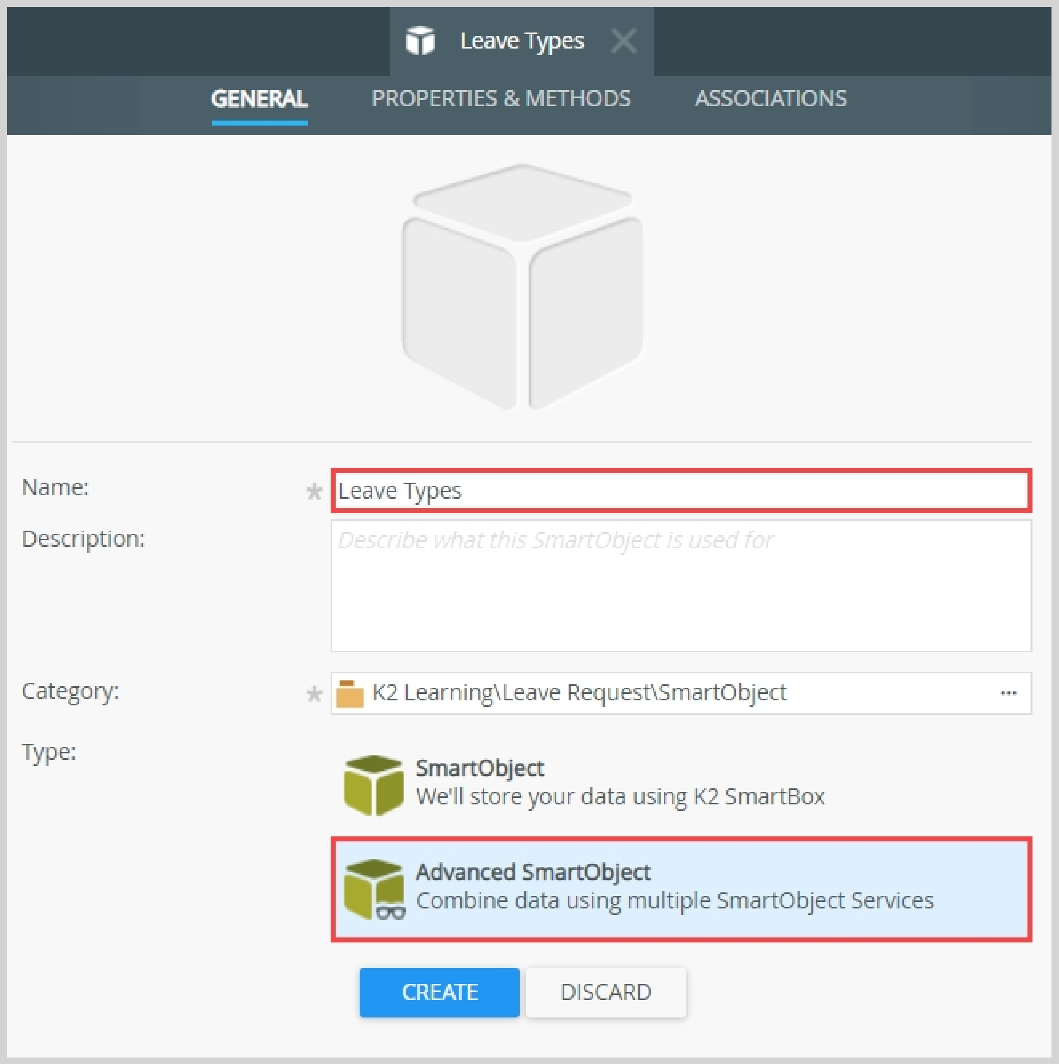
In this step, you created a new service instance based on the SQL Server Service type. The new service instance contains the configuration necessary to connect to an external SQL database.

Next Step: 3. Create a new SmartObject from a Service Instance

3. Create a new SmartObject from a Service Instance

In this step, you create a new SmartObject from the Leave Types service instance. You will add the properties discovered in the service instance, along with one method, List. The list method returns a list of leave types that you will use in your Leave Request Form.

1. Return to **K2 Designer**. From the **Leave Request** > **SmartObjects** category, create a new SmartObject called *Leave Types*. Add the **List** method from the **Leave Types** service instance. **Create** new SmartObject properties for the LeaveTypeId and LeaveTypeDescription data source properties.
    a. In this step, you manually create a new SmartObject from the **Leave Types** service instance. Return to **K2 Designer**, then right-click the **Leave Request** > **SmartObjects** category and select **New SmartObject**.
    b. Name the SmartObject
    Leave Types
    then select **Advanced SmartObject**. Use the advanced option when creating SmartObjects from a service instance. Click **Create**.



Name: * Leave Types

Description: Describe what this SmartObject is used for

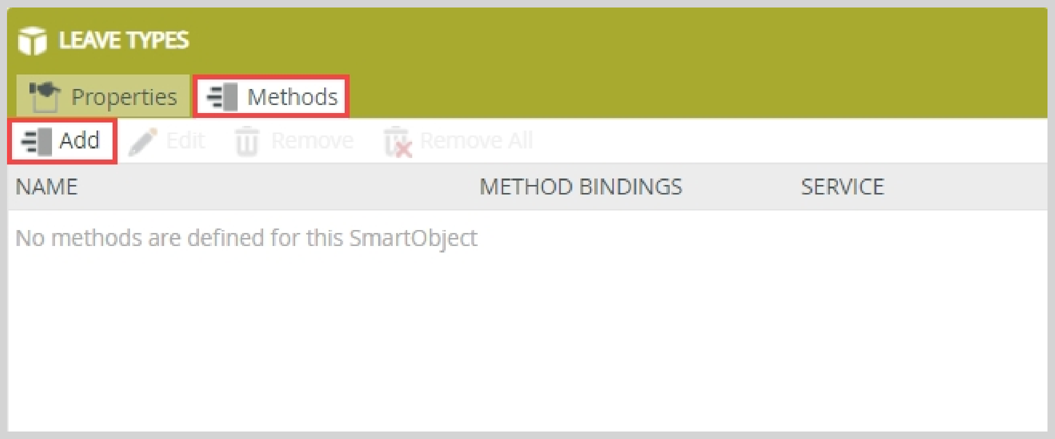
Category: * K2 Learning\Leave Request\SmartObject

Type:

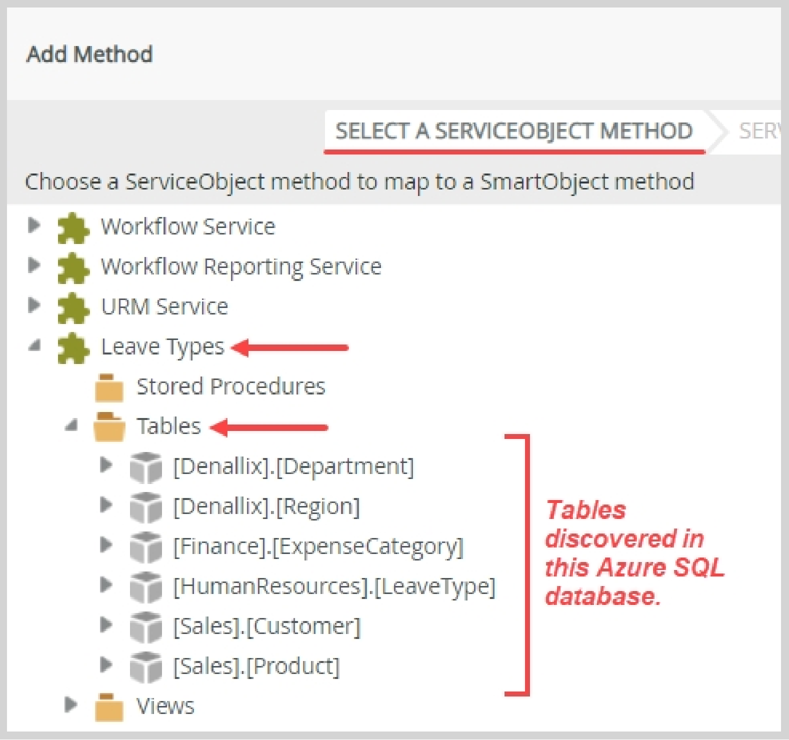
SmartObject
We'll store your data using K2 SmartBox

Advanced SmartObject
Combine data using multiple SmartObject Services

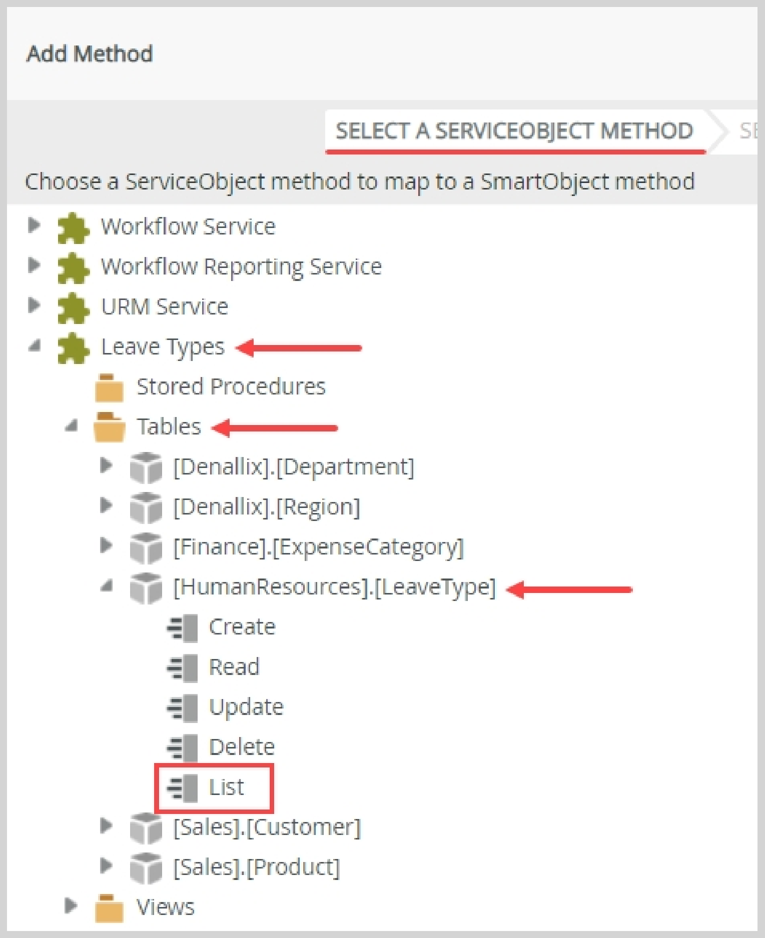
c. Next, you will bind the SmartObject to the Leave Types service instance. Click the **Methods** tab. You want to add only one method, List. You will see there are many methods and you could add all of them if you needed. This is one advantage to creating your SmartObject manually - you can pick and choose the properties and methods you want. Click the **Add** button.



d. On the **Select a ServiceObject Method** screen, locate the **Leave Types** service instance. Expand the **Leave Types** node, then expand the **Tables** node. There are six tables in this particular database that K2 discovered when you created the service instance.


Add Method
SELECT A SERVICEOBJECT METHOD
SER
Choose a ServiceObject method to map to a SmartObject method
Workflow Service
Workflow Reporting Service
URM Service
Leave Types
Stored Procedures
Tables
[Denallix].[Department]
[Denallix].[Region]
[Finance].[ExpenseCategory]
[HumanResources].[LeaveType]
[Sales].[Customer]
[Sales].[Product]
Views
Tables discovered in this Azure SQL database.

e. Expand the **[HumanResources].[LeaveType]** table. Here, you see a list of all the methods discovered for this table. Click **List**, then click **Next**.



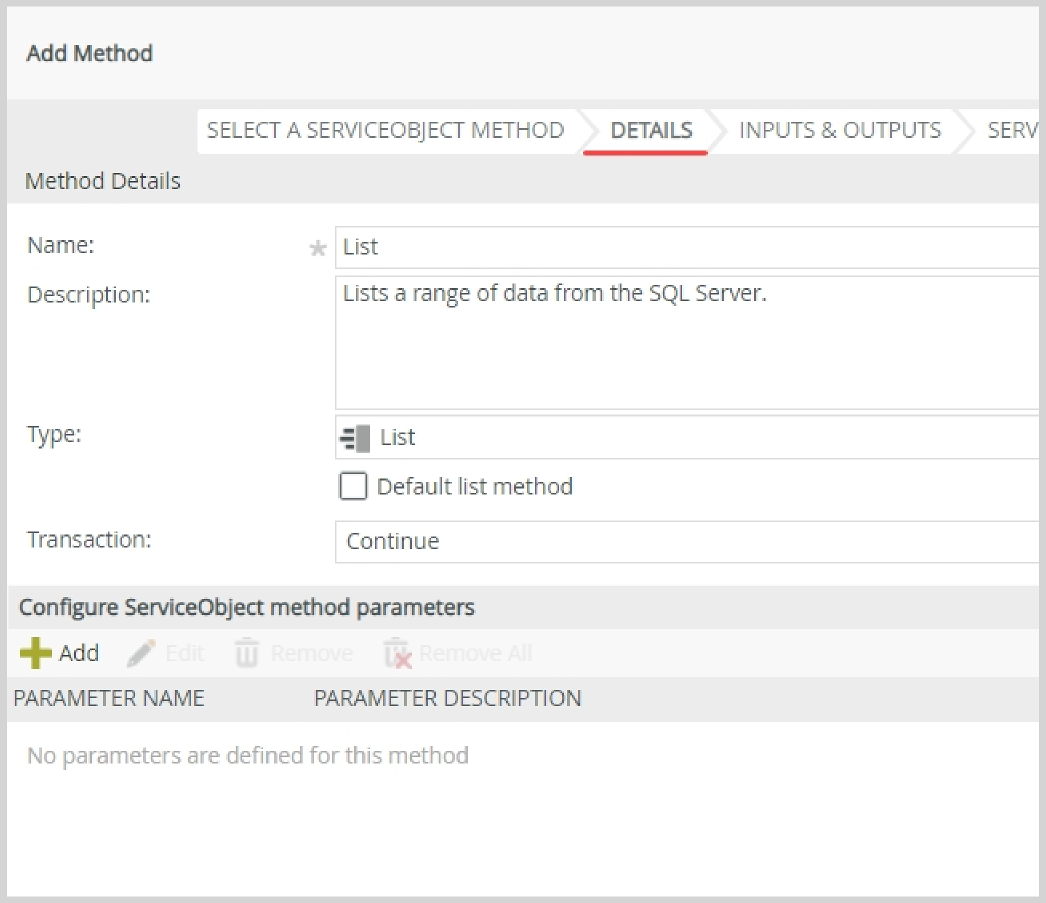
Add Method

SELECT A SERVICEOBJECT METHOD

Choose a ServiceObject method to map to a SmartObject method

- Workflow Service
- Workflow Reporting Service
- URM Service
- Leave Types
  - Stored Procedures
  - Tables
    - [Denallix].[Department]
    - [Denallix].[Region]
    - [Finance].[ExpenseCategory]
    - [HumanResources].[LeaveType]
      - Create
      - Read
      - Update
      - Delete
      - List
    - [Sales].[Customer]
    - [Sales].[Product]
  - Views

f. You are now on the **Details** screen. This screen gives you a description of what the List method does, and you can configure parameters if you need too. (Think of parameters as pre-set values the method requires to run. In this case, you do not need any parameters for the List method.) Click **Next**.



Add Method

Method Details

Name: List

Description: Lists a range of data from the SQL Server.

Type: List

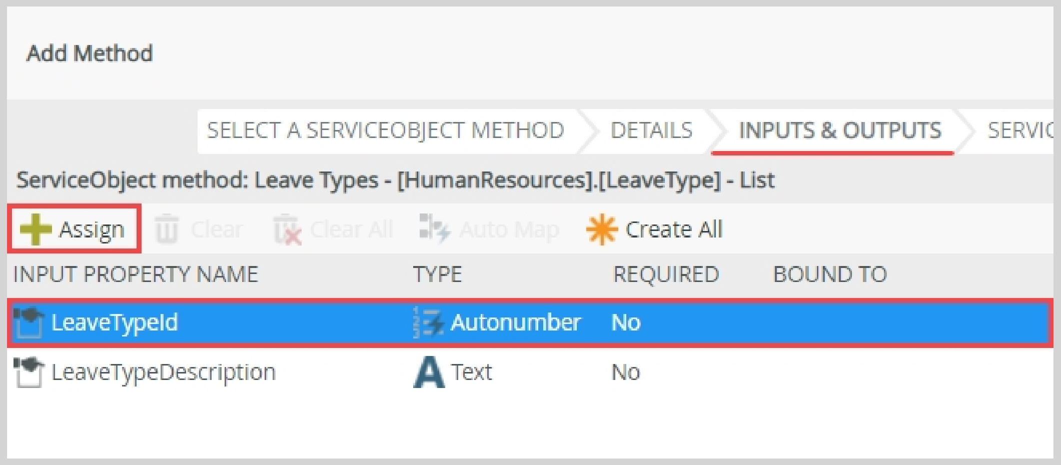
□ Default list method

Transaction: Continue

Configure ServiceObject method parameters

PARAMETER NAME	PARAMETER DESCRIPTION
No parameters are defined for this method	

g. You are now on the **Inputs & Outputs** screen. Notice the two values under the input section: **LeaveTypeId** and **LeaveTypeDescription**. These are the properties K2 discovered in the data source. In other words, these are the columns in the **[HumanResources].[LeaveType]** table. You want to create matching properties in your new SmartObject. You are creating an association in your SmartObject properties to the corresponding properties in the data source. Select the **LeaveTypeId** row, then click **Assign**.

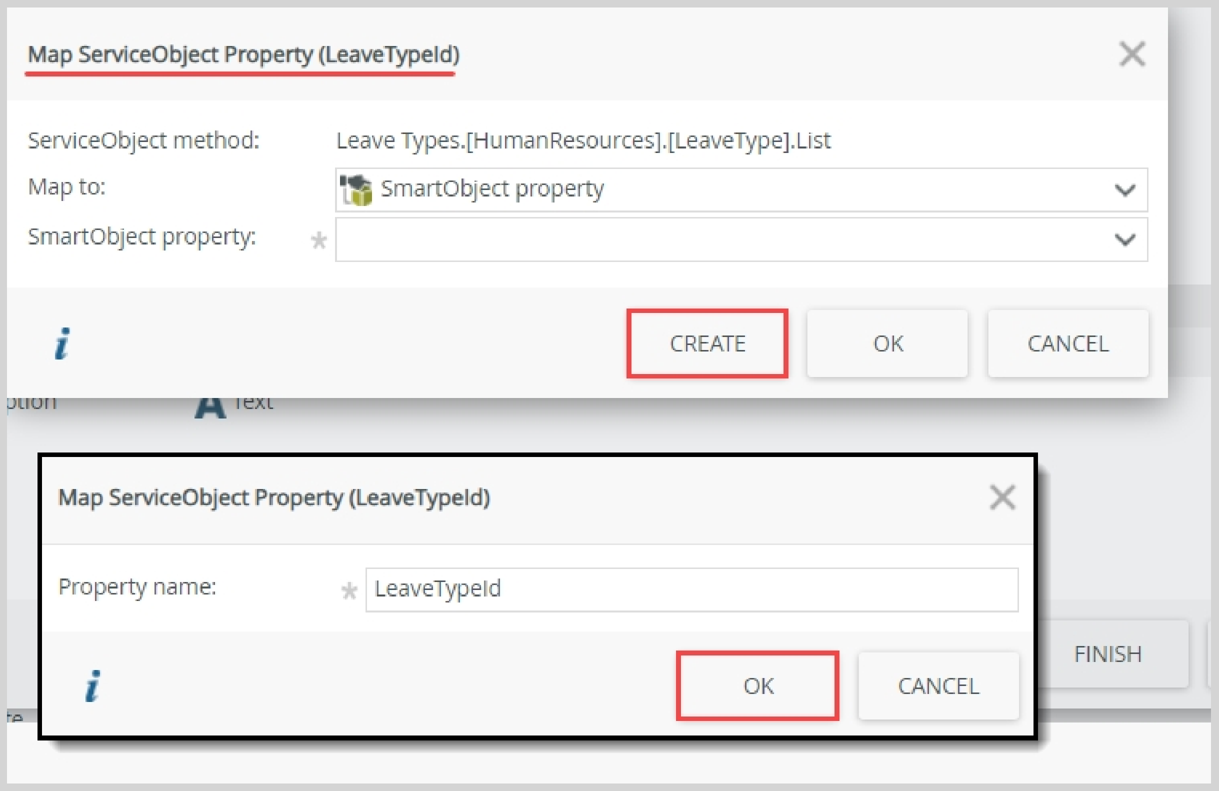


Add Method

ServiceObject method: Leave Types - [HumanResources].[LeaveType] - List

INPUT PROPERTY NAME	TYPE	REQUIRED	BOUND TO
LeaveTypeId	Autonumber	No	
LeaveTypeDescription	Text	No	

h. You don't have an existing property to map in your new SmartObject, so you need to create a new one. Click the **Create** button. Notice K2 matches the new property name with the existing service instance property name. Click **OK**, then click **OK** again.



Map ServiceObject Property (LeaveTypeId)

ServiceObject method: Leave Types.[HumanResources].[LeaveType].List

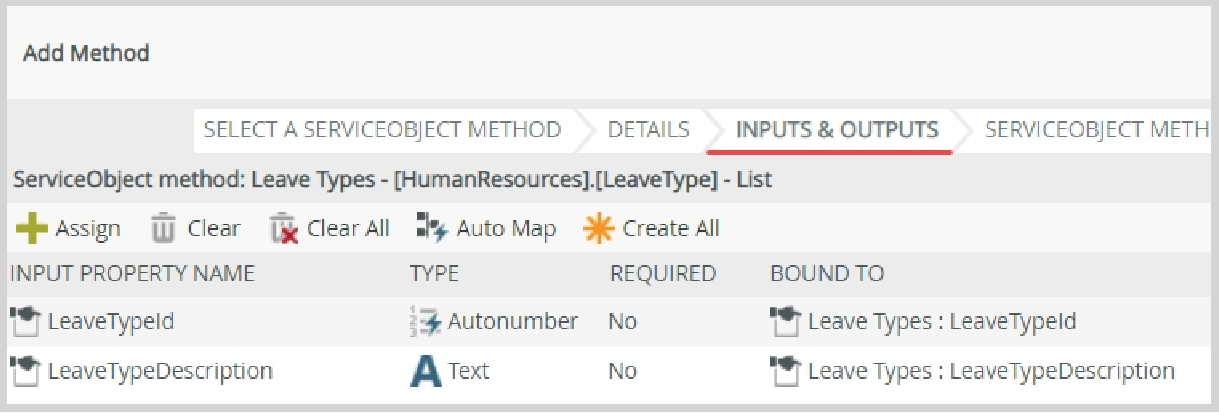
Map to: SmartObject property

SmartObject property: *

Map ServiceObject Property (LeaveTypeId)

Property name: * LeaveTypeId

i. Repeat the steps above to create a new property for **LeaveTypeDescription**. You should now have both service instance properties bound to properties in your new SmartObject. You don't need to edit the return properties since you are just retrieving leave types. Click **Next**, then **Finish** to complete this method configuration.

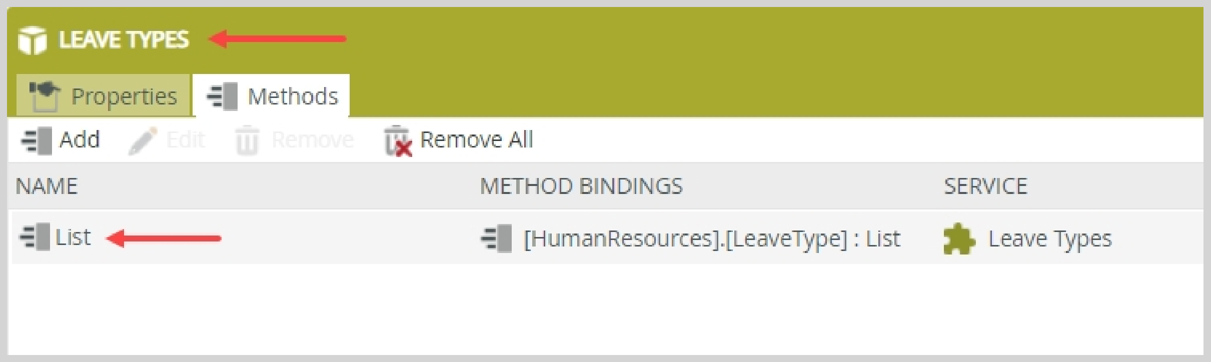


Add Method

ServiceObject method: Leave Types - [HumanResources].[LeaveType] - List

INPUT PROPERTY NAME	TYPE	REQUIRED	BOUND TO
LeaveTypeId	Autonumber	No	Leave Types : LeaveTypeId
LeaveTypeDescription	Text	No	Leave Types : LeaveTypeDescription

j. Your new Leave Types SmartObject has one method, **List**. Click **Finish** to save and exit the SmartObject.



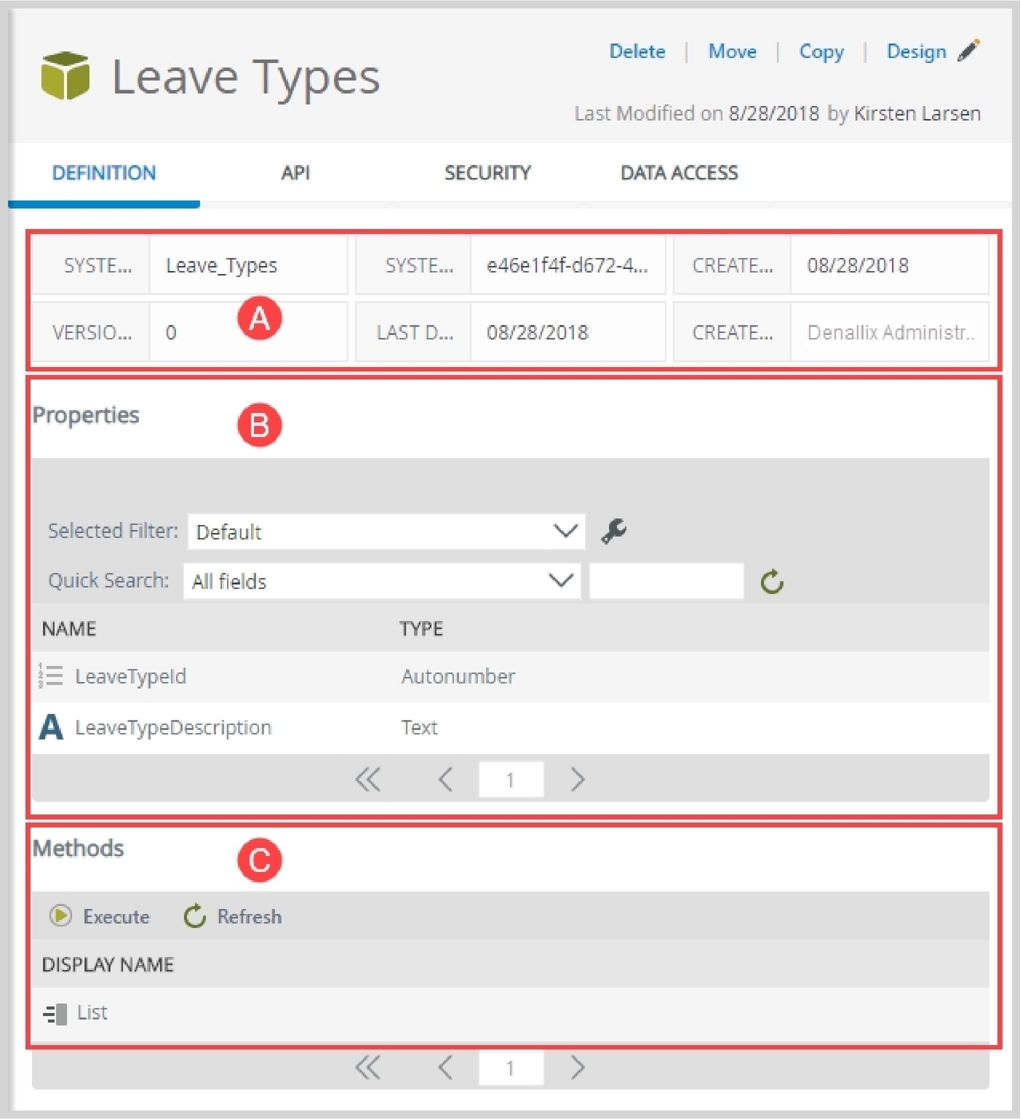
2. Before moving on, execute the **List** method for the **Leave Types SmartObject** in the **K2 Management** site. Confirm results are returned with two properties shown: **LeaveTypeId** and **LeaveTypeDescription**. You do not need to enter any input properties, just let K2 return all the results it finds.

   a. Next, you will test the SmartObject configuration by executing the list method. The list method returns all the records found in the data source. You can enter input properties to filter the results if necessary, but for this test, you will return all records.
   Return to the **K2 Management** site. Expand the **Leave Request** > **SmartObjects** categories. You should see the two SmartObjects created for this application: *Leave Request SmartObject* and *Leave Types*. (You may need to right-click the category and refresh the menu if you don't see both SmartObjects.)

   b. Select the **Leave Types** SmartObject. The details open in the central pane. Take a minute to review the following sections:

      - **A**: Displays the system details such as the system name, date created, and created by values.
      - **B**: Displays the properties: LeaveTypeDescription and LeaveTypeId (think column titles in a spreadsheet).
      - **C**: Displays the methods for this table. You should only see the List method as you only added it to the SmartObject configuration. The List method returns a list of leave types for the Leave Type drop-down list on the item view. Put simply, methods are how K2 interacts with data: create, save,

delete, load, list.



Leave Types

Last Modified on 8/28/2018 by Kirsten Larsen

SYSTE...	Leave_Types	SYSTE...	e46e1f4f-d672-4...	CREATE...	08/28/2018
VERSIO...	0	LAST D...	08/28/2018	CREATE...	Denallix Administr..

Properties

Selected Filter: Default

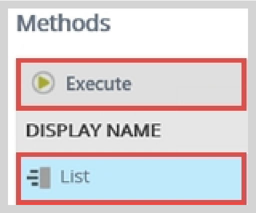
Quick Search: All fields

NAME	TYPE
LeaveTypeId	Autonumber
LeaveTypeDescription	Text

Methods

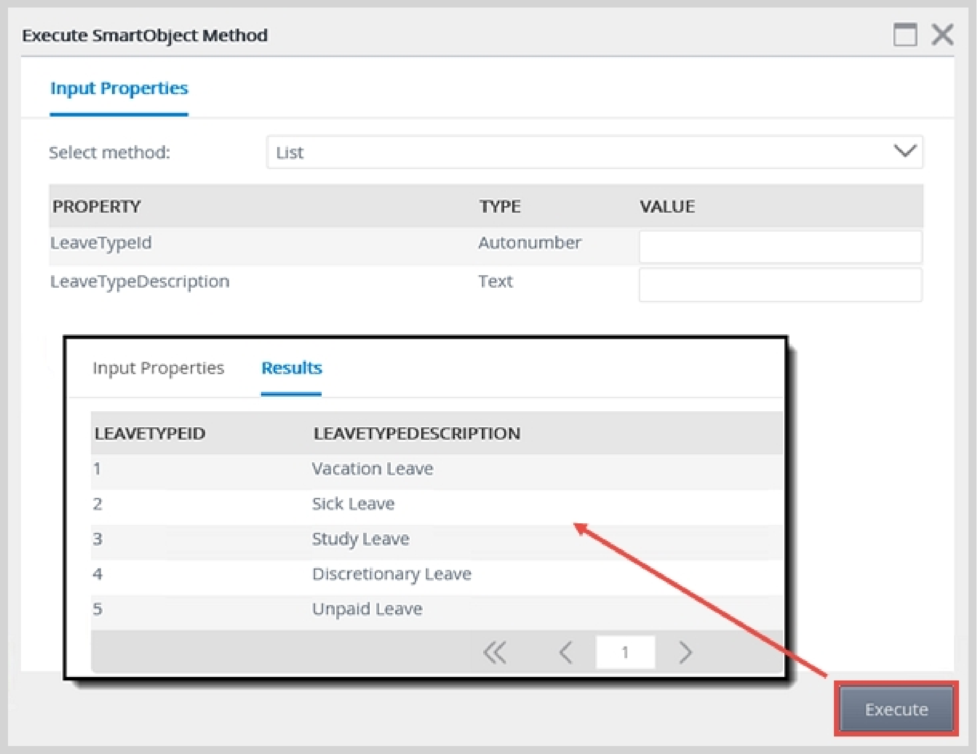
DISPLAY NAME
List

c. In the **Methods** pane, select the **List** method, then click **Execute**.



d. There are no input properties for this test, so click **Execute** once again. The Results pane appears with the values from the Leave Types SmartObject. Click **Done** to exit the **Execute SmartObject**

Method screen.



Select method: List

PROPERTY	TYPE	VALUE
LeaveTypeId	Autonumber	
LeaveTypeDescription	Text	

LEAVETYPEID	LEAVETYPEDESCRIPTION
1	Vacation Leave
2	Sick Leave
3	Study Leave
4	Discretionary Leave
5	Unpaid Leave

Review

In this step, you created a new SmartObject based on the Leave Types service instance. You added the List method only, because you do not need the other methods for this application. You created properties in your new SmartObject that are bound to the properties in the service instance. Finally, you tested the new SmartObject configuration by executing the list method and confirming there were leave type results returned. In the next step, you add the new Approver Comments to the item view.

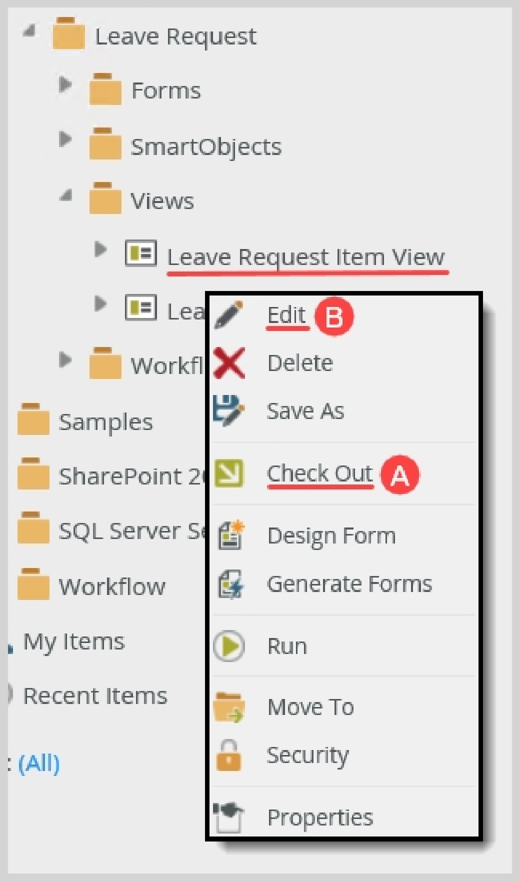
Next Step: 4. Add the Approver Comments Control to the Leave Request Item View and Bind the Leave Type Drop-Down List

4. Add the Approver Comments Control to the Leave Request Item View and Bind the Leave Type Drop-Down List

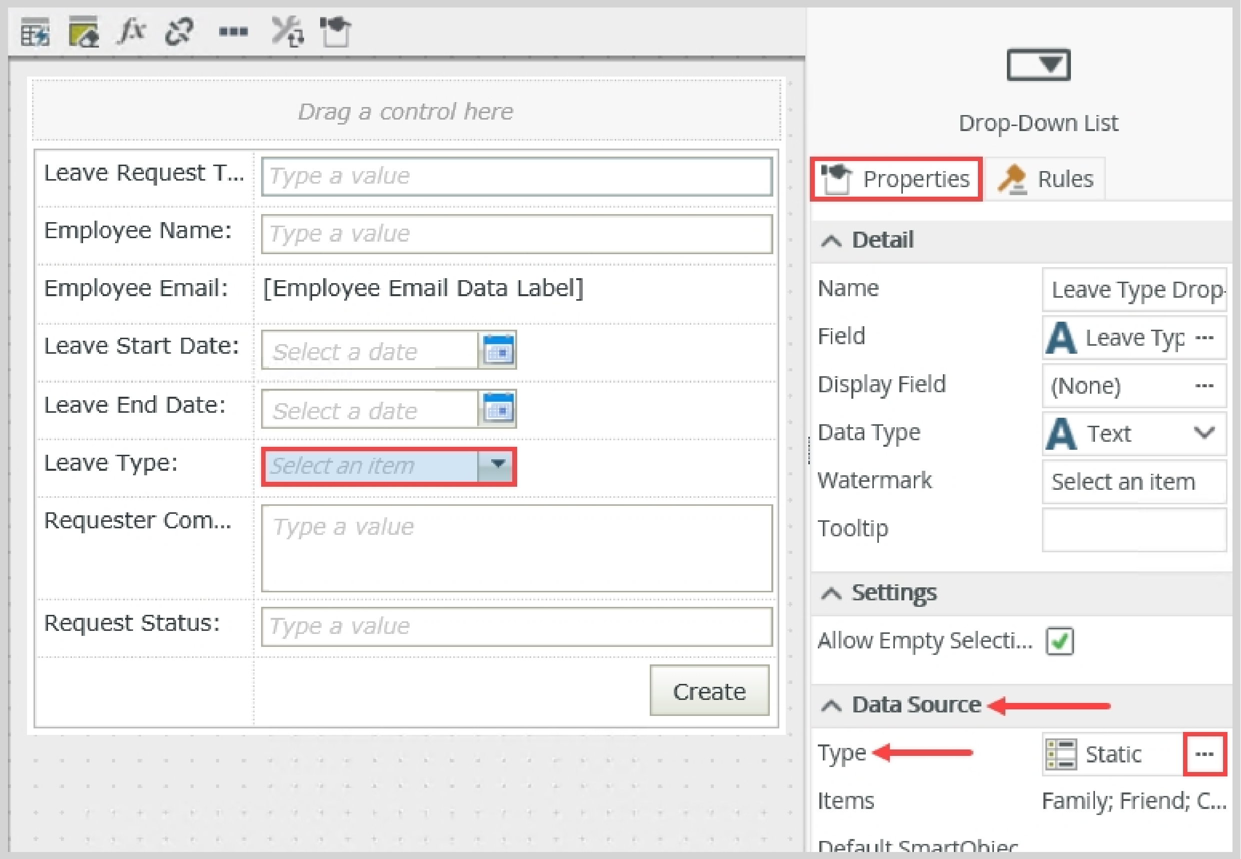
In this step, you will add a new row to the layout table for the Leave Request Item View, and add the new Approver Comments property to the view. You will also "bind" the Leave Type drop-down list to the Leave Types SmartObject. (In the basic version tutorial, you entered static leave type values for the drop-down list. After binding the control to the Leave Types SmartObject, the control retrieves its values dynamically from the Azure SQL database when the form loads.)

You will also delete the Request Status control and label. Since the workflow updates this property, it may be confusing to the user to see this unused field on their Leave Request Form.

1. Returning to the K2 Designer, **edit** the **Leave Request Item View** and change the data source for the **Leave Type** drop-down list from static to the Leave Types SmartObject. Make the **LeaveTypeDescription** property the control's **Display** and **Value**. Here, you are changing the drop-down control from using static list item values to using dynamic values that K2 retrieves from the Azure SQL Database.
    a. In this section, you will edit the data source properties for the Leave Type drop-down list. You will bind the control to the Leave Types SmartObject.
    Because you are changing the view, you must check it out first. Switch back to the K2 Designer (it should be a tab in your browser). Right-click and check out the **Leave Request Item View**, then right-click and select **Edit**.



b. Next, you will bind the drop-down list to the Leave Types SmartObject. Click the **Leave Type** drop-down list to highlight it. (If you see a dialog indicating the view is not checked out, click **Yes** to check out the view and continue.) In the **Properties** pane, open the data type editor by clicking the **Data Source** > **Type** ellipsis.



Leave Request T...	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Com...	Type a value
Request Status:	Type a value

Detail

Name	Leave Type Drop-
Field	Leave Typ
Display Field	(None)
Data Type	Text
Watermark	Select an item
Tooltip	

Settings

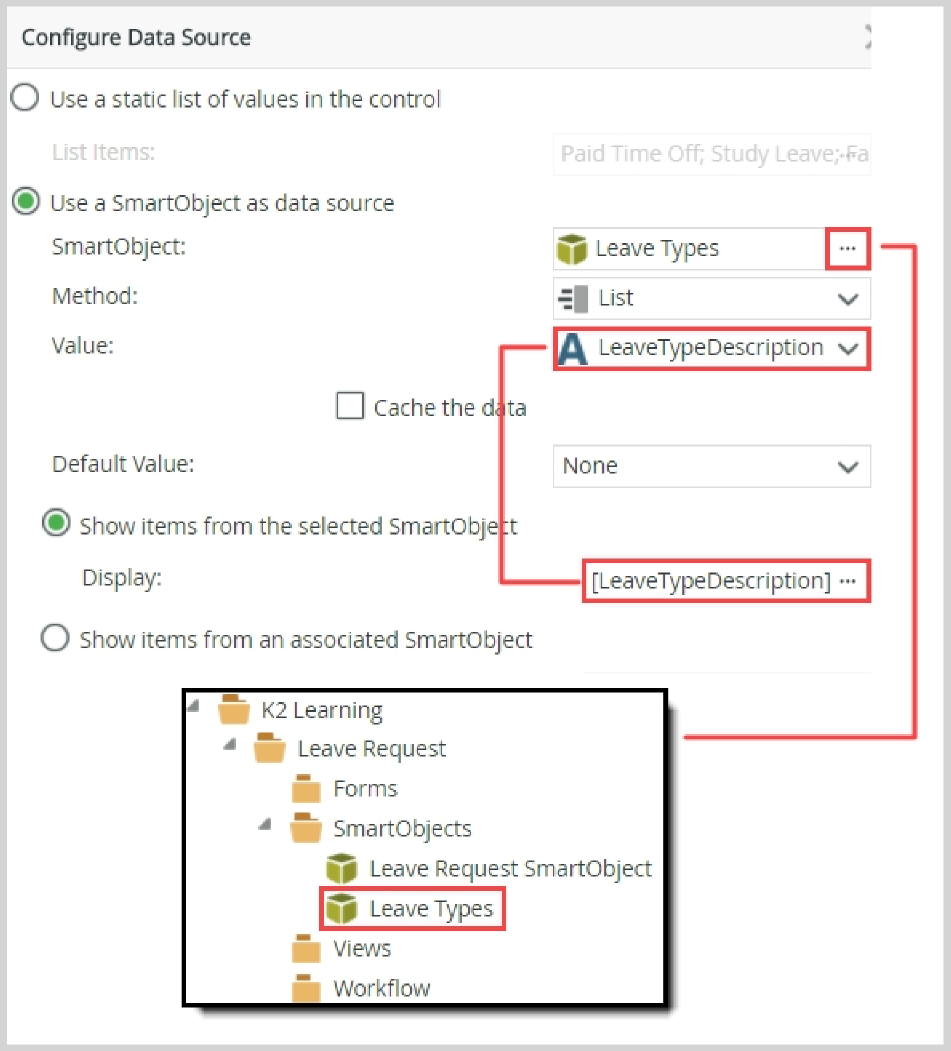
Allow Empty Selecti...	☑

Data Source

Type	Static
Items	Family; Friend; C...
Default SmartObjec	

- Select the **Use a SmartObject as data source** option, then click the SmartObject ellipsis.
- Navigate to, then select, the **Leave Types** SmartObject that you created in the last tutorial. Click **OK**.

- Change the **Value** to the **LeaveTypeDescription** property so that it matches the **Display**.



Configure Data Source

○ Use a static list of values in the control

List Items: Paid Time Off; Study Leave;-Fa

● Use a SmartObject as data source

SmartObject: Leave Types ···

Method: List

Value: LeaveTypeDescription

□ Cache the data

Default Value: None

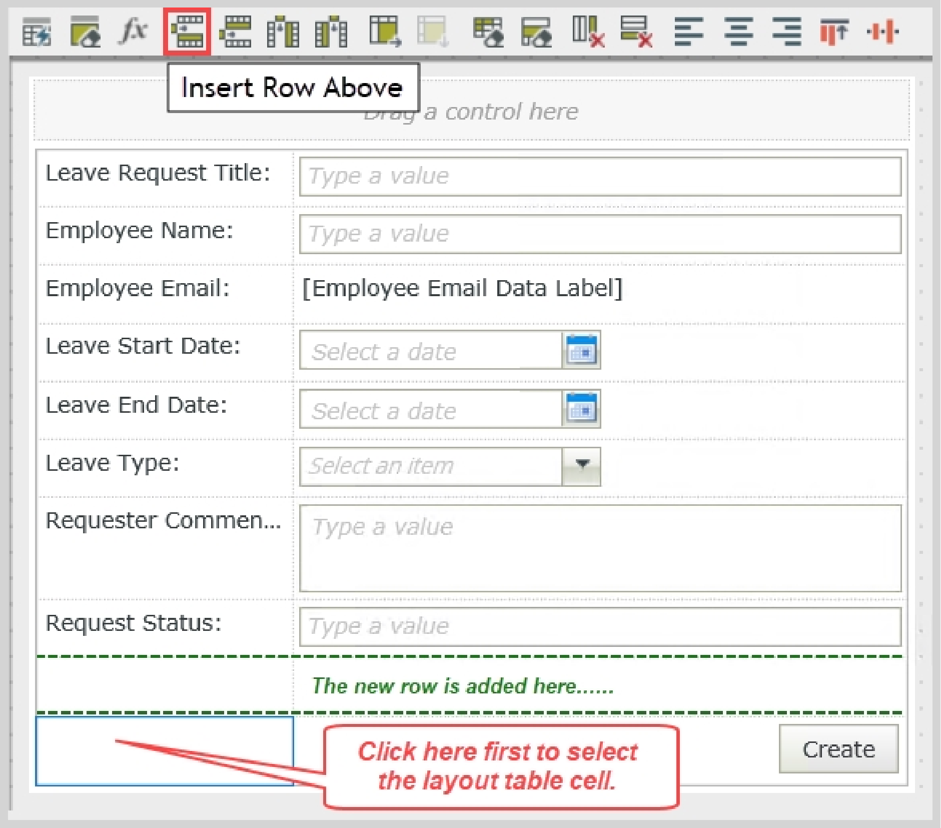
● Show items from the selected SmartObject

Display: [LeaveTypeDescription] ···

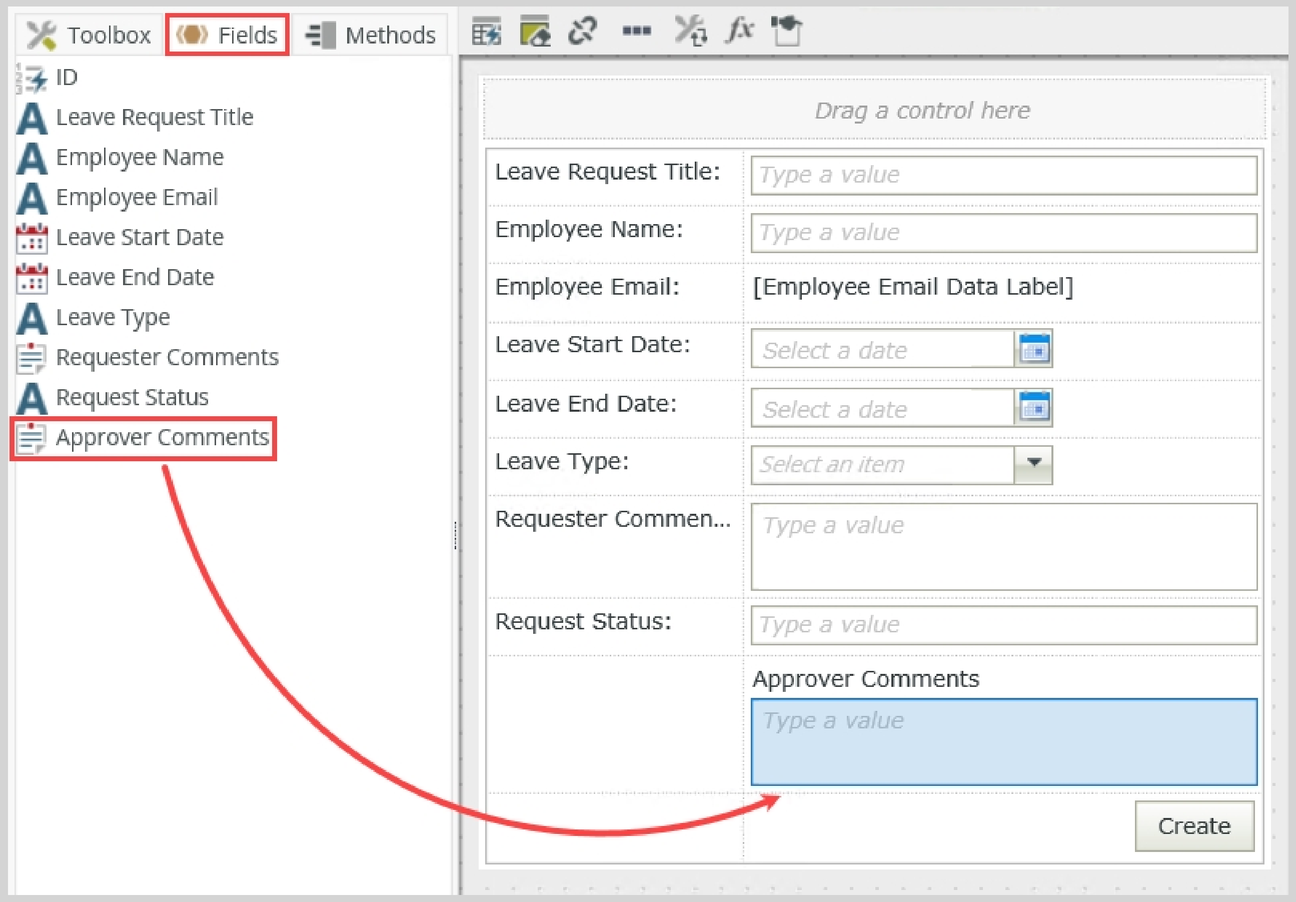
○ Show items from an associated SmartObject

- Click **OK** to return to the design canvas.

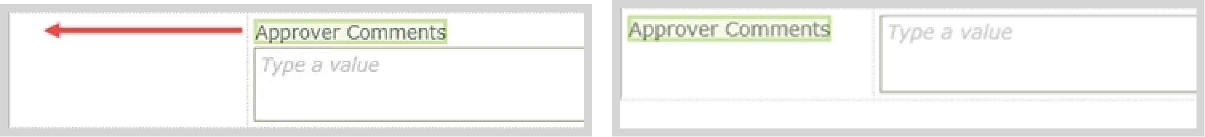
2. Add a new row to the view layout just above the **Create** button row. Add the **Approver Comments** field to the new row, moving the label to the first cell and the field to the second cell. Make the field **read-only** by default. In a later step, you will edit a state and enable this control for the approving manager's task. This allows the manager to enter any questions or comments they may have about the leave request.
   a. In the next few steps, you will edit the view layout and add the new **Approver Comments** property from the Leave Request SmartObject. You will make the control read-only by default, because the leave requester has their own field for entering comments. This field is for the approving manager only. In a later step, you will edit a rule and enable this control for the approving manager.
   Begin by adding a new row to the view layout table. Click once in the bottom row, first cell, to highlight the cell. Click the **Insert Row Above** icon found in the view canvas menu.



b. Click the **Fields** tab in the left-side column to expose the SmartObject properties for this view. Drag the **Approver Comments** property into the new row, second cell.

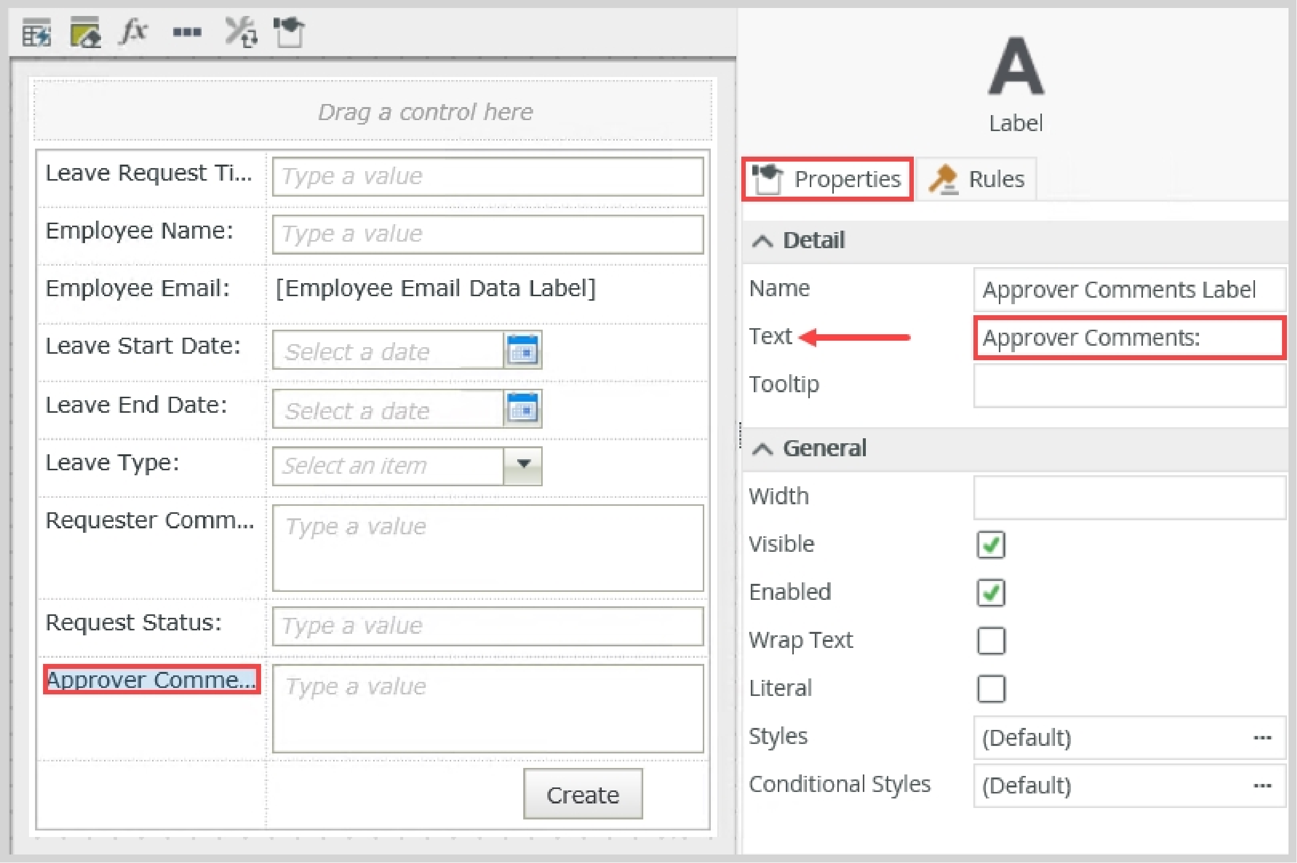


c. Notice that K2 places the **Approver Comments** label above the text area control. Click and drag the **label** into the empty cell to the left.



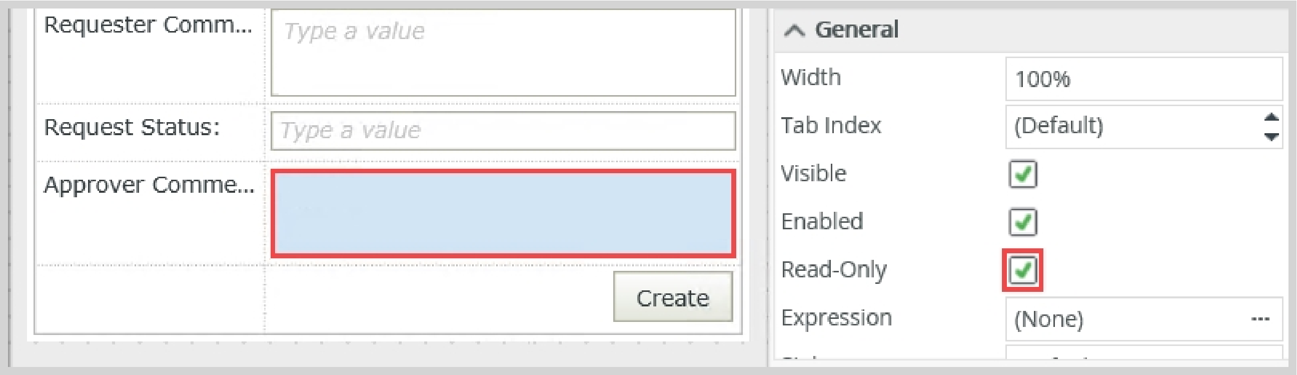
d. There are two minor adjustments to make to the new label and text area control. First, highlight the label, then locate the **Text** value in the Properties pane. Add a colon (:) to the end of the text value so that it matches the other labels in the view.

> **Note**
> Notice the **Name** and the **Text** properties in the **Properties** pane. The **Name** value is the control's identity that K2 uses to reference a specific control. For example, in rules. In this case, the name value is Approver Comments Label. The **Text** value is what the user sees when they view the form.



Name: Approver Comments Label
Text: Approver Comments:
Tooltip:
Width:
Visible: ☑
Enabled: ☑
Wrap Text: □
Literal: □
Styles: (Default) ...
Conditional Styles: (Default) ...

e. Highlight the text area control. In the Properties pane, CHECK the **Read-Only** option. Since this control is for any questions or comments the approving manager may have, you don't want the requester to edit it.

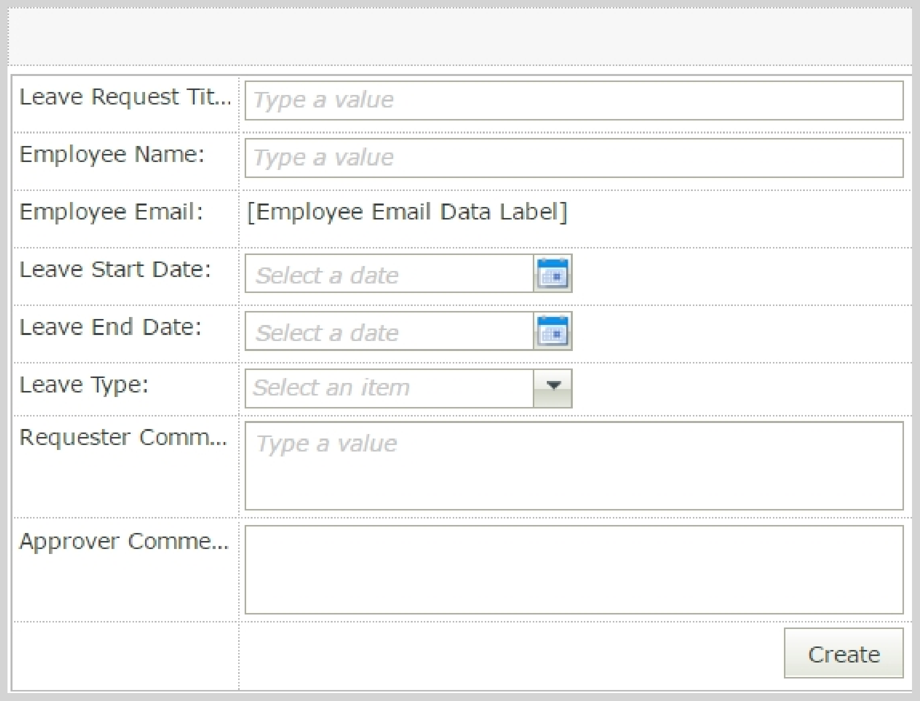


Requester Comm...	Type a value
Request Status:	Type a value
Approver Comme...	
	Create

General

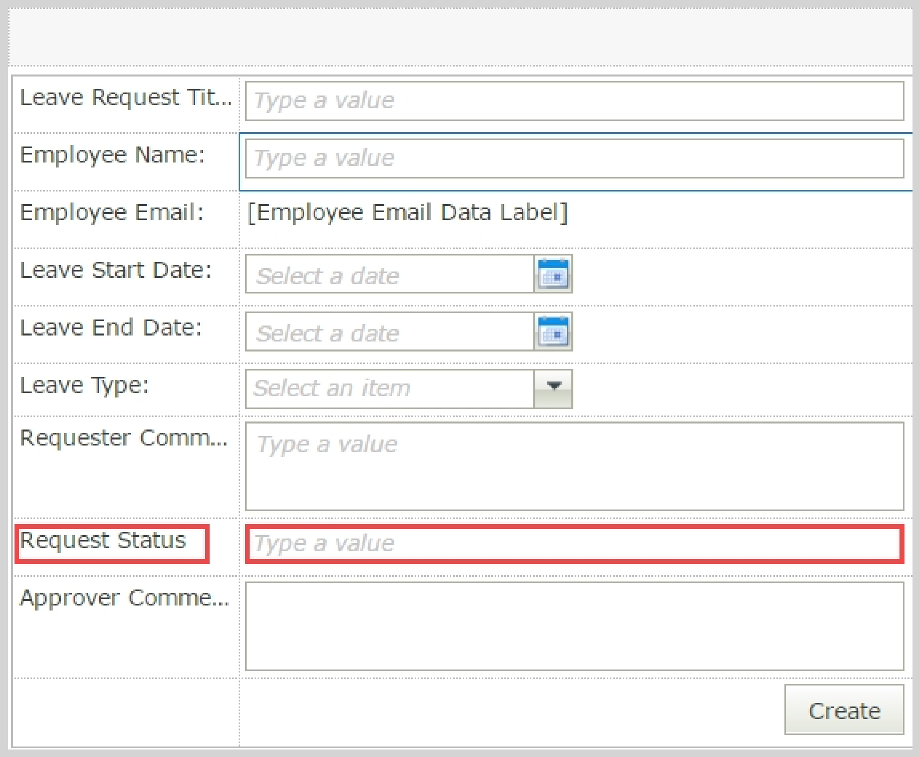
Width	100%
Tab Index	(Default)
Visible	☑
Enabled	☑
Read-Only	☑
Expression	(None) ···

3. In the last step for this section, you will remove a control and label that may be confusing to your users. The workflow updates the status property in the Leave Request SmartObject. Since neither the requester or the approving manager need to edit this field, you will remove it from the view.
   Remove the **Request Status** label and text box. When you see the warning about dependencies, select the option to **Remove all dependencies**.



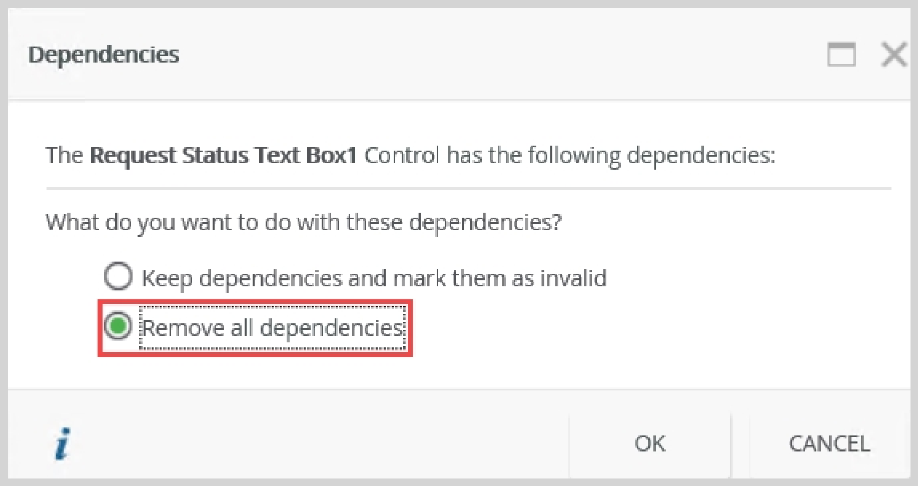
Leave Request Tit...	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comm...	Type a value
Approver Comme...	
	Create

   a. Next, you will delete the Request Status label and text box control. The workflow updates this SmartObject property behind-the-scenes. It may confusing to have it displayed, when there is no user input necessary.
      Select the **Request Status Label** to highlight it, then click the <DELETE> button (on your keyboard).



Leave Request Tit...	Type a value
Employee Name:	Type a value
Employee Email:	[Employee Email Data Label]
Leave Start Date:	Select a date
Leave End Date:	Select a date
Leave Type:	Select an item
Requester Comm...	Type a value
Request Status	Type a value
Approver Comme...	
	Create

b. Repeat this process for the **Request Status Text Box**. You will see a warning about dependencies, click the **Remove all dependencies** option.



Dependencies

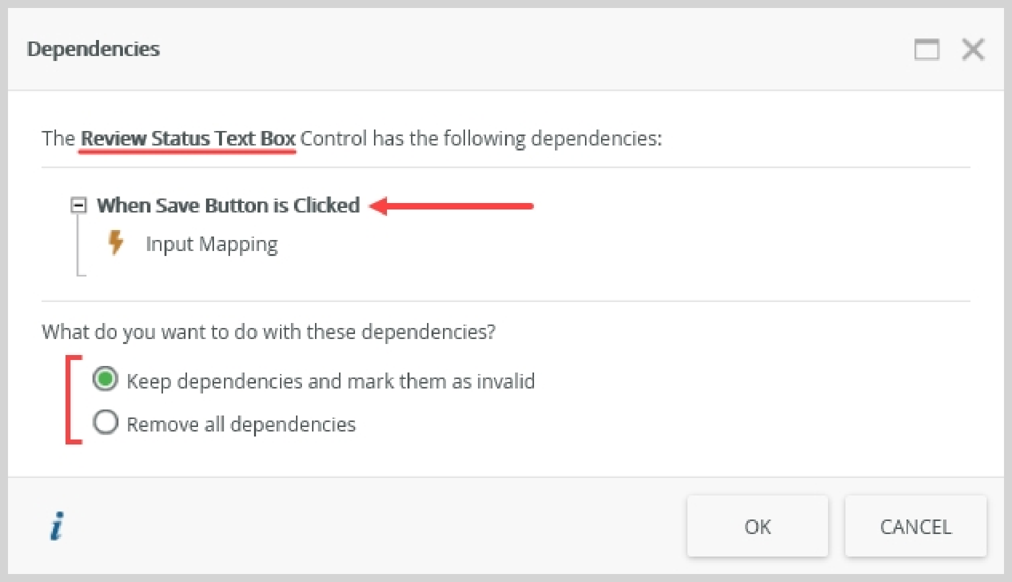
The **Request Status Text Box1** Control has the following dependencies:

What do you want to do with these dependencies?

○ Keep dependencies and mark them as invalid

● Remove all dependencies

Note
When you change a control that results in a rule discrepancy, you will see the **Dependencies** warning. At the top of the dialog, you will see the reference control name. The middle pane displays the rules that are dependent on the control. In the sample below, the **Request Status Text Box** control is being deleted from the view, resulting in the dependencies warning. There is one rule dependent on the Request Status Text Box control: **When Save Button is Clicked**.



Dependencies

The Review Status Text Box Control has the following dependencies:

When Save Button is Clicked
- Input Mapping

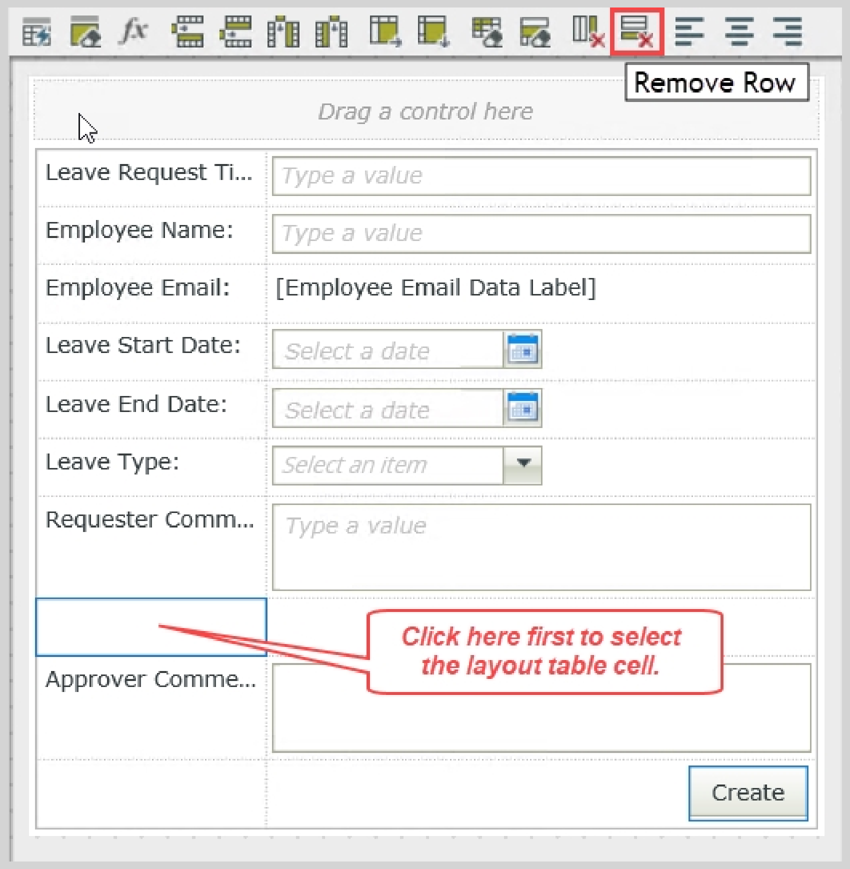
What do you want to do with these dependencies?

- ● Keep dependencies and mark them as invalid
- ○ Remove all dependencies

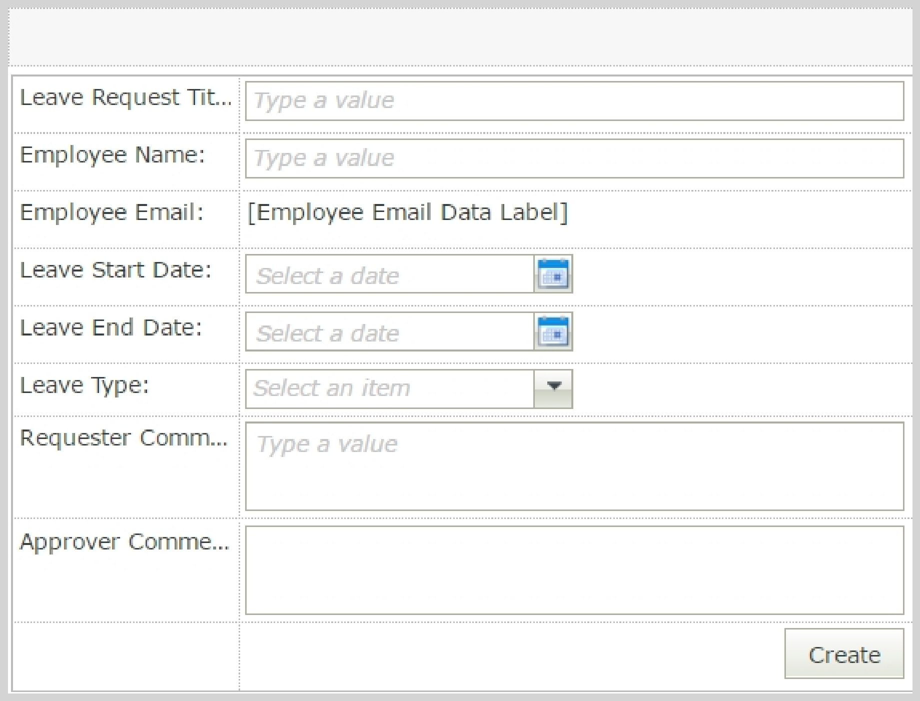
You have two options:

- **Keep dependencies and mark them as invalid**: This option keeps the dependency in place and flags its location (in the rule) with a red error badge. You must edit the rule and fix or remove the dependency. You cannot check in a view or form that has dependency errors. In this example, deleting the Request Status Text Box control results in the Create Button input mapping property becoming invalid. Since you deleted the control from the view, the fix here is to delete the control reference from input mappings.
- **Remove all dependencies**: This option removes the dependency from any rules that reference it.

c. You can delete the table row if you wish using the **Remove Row** icon from the view canvas menu. To do that, click an empty cell in the row you want to remove, then click the **Remove Row** icon. Click **OK** to continue.

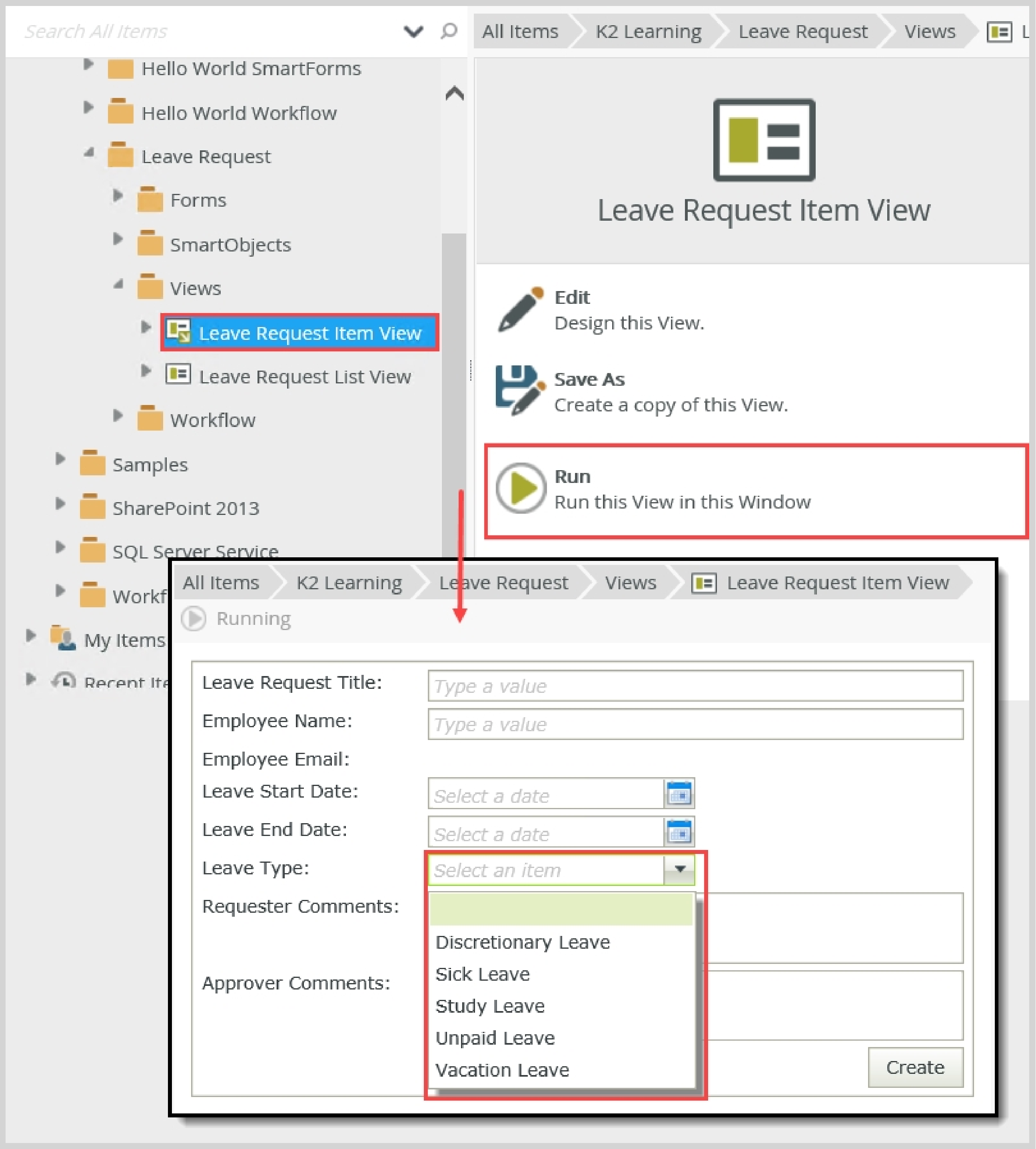


Click here first to select the layout table cell.



d. Click **FINISH** to save and exit the view.

e. Select the view title in the explorer (if it is not already) then click the **Run** button found in the central pane. Confirm the **Leave Type** drop-down list now contains the values from the Leave Types SmartObject. In this step, you are simply testing the SmartObject connection for the Leave Type drop-down list.



Leave Request Title: Type a value

Employee Name: Type a value

Employee Email:

Leave Start Date: Select a date

Leave End Date: Select a date

Leave Type: Select an item

- Discretionary Leave
- Sick Leave
- Study Leave
- Unpaid Leave
- Vacation Leave

Requester Comments:

Approver Comments:

f. Click once again on the **Leave Request Item View** title to close the Run screen.

Review

In this step, you edited the Leave Request Item View. You changed the data source for the Leave Type drop-down list control from static values to dynamic values. This control now retrieves its list items from the Leave Types SmartObject. Using a SmartObject for list items has several advantages, one of which is the ability to edit the list items in one place. If you use this list for several list controls, the list items will be consistent for all controls.

You also added the Approver Comments property to the view and made it read-only so that the requester cannot edit the values. Later on, you will enable the Approver Comments control so that the approving manager can enter any questions or comments they may have.

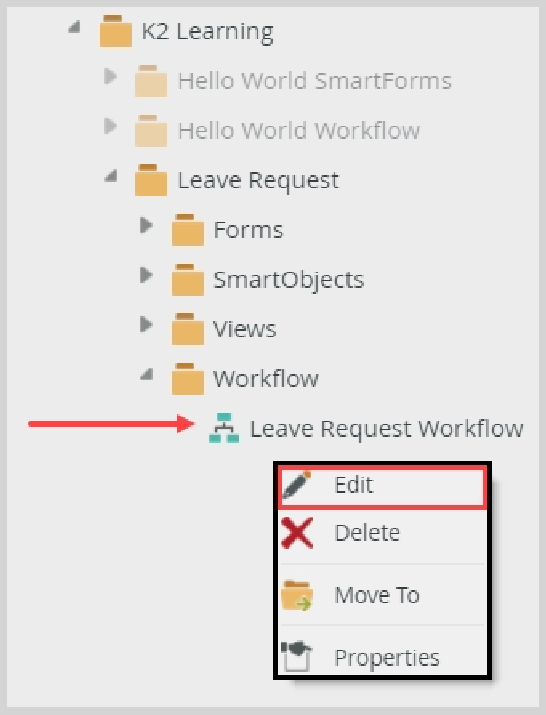
In the next step, you will edit the Leave Request Workflow. Some of the changes you will make include adding a rework loop and adding Send Email steps.

Next Step: 5. Add and Configure Email Steps for the Manager Outcomes

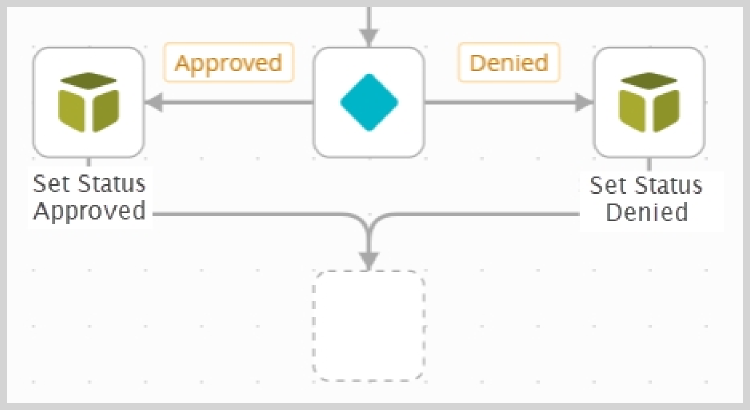
5. Add and Configure Email Steps for the Manager Outcomes

In this step you will expand the Leave Request Workflow by adding email steps. You will add a Send Email step to the approved and denied outcomes and direct them to the leave requester. This will let the requester know the manager's decision.

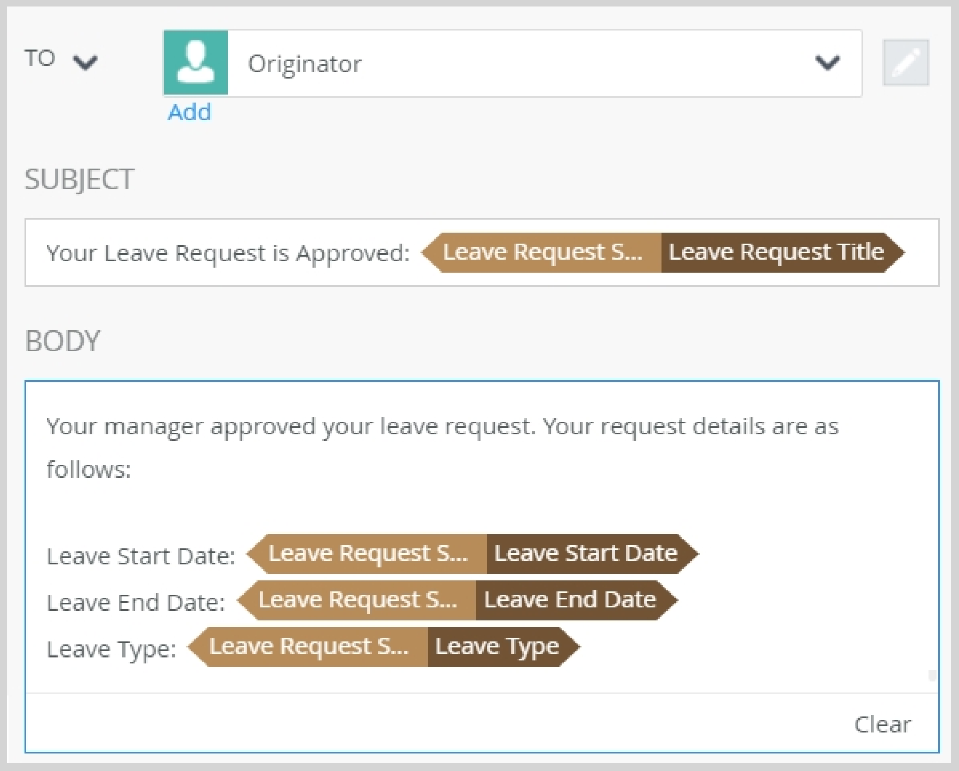
1. **Edit** the Leave Request Workflow. Begin by deleting the **End** step. The two outcome lines will still connect to an empty placeholder. You will add email steps, then reroute the outcomes to the new steps.
   a. In K2 Designer, edit the **Leave Request Workflow**. Right-click the workflow name in the explorer, then select **Edit**. When you see the helpful tips dialog, click **CLOSE**.



   b. Select the **End** step, then click <DELETE> on your keyboard. The outcomes still connect to an empty placeholder. You will delete the placeholder in a later step.



2. Add a **Send Email** step and align it with the Approved outcome. Connect the Approved outcome to the email step.
   - In the Configuration Panel, configure the email **recipient** as the **Originator**. Customize the SUBJECT line and message BODY using references from the **Leave Request SmartObject**. Use the image below for reference, if needed. To access SmartObject properties, expand the Context Browser. Under the REFERENCE heading, expand the Leave Request SmartObject. Drag the SmartObject references into the message body.



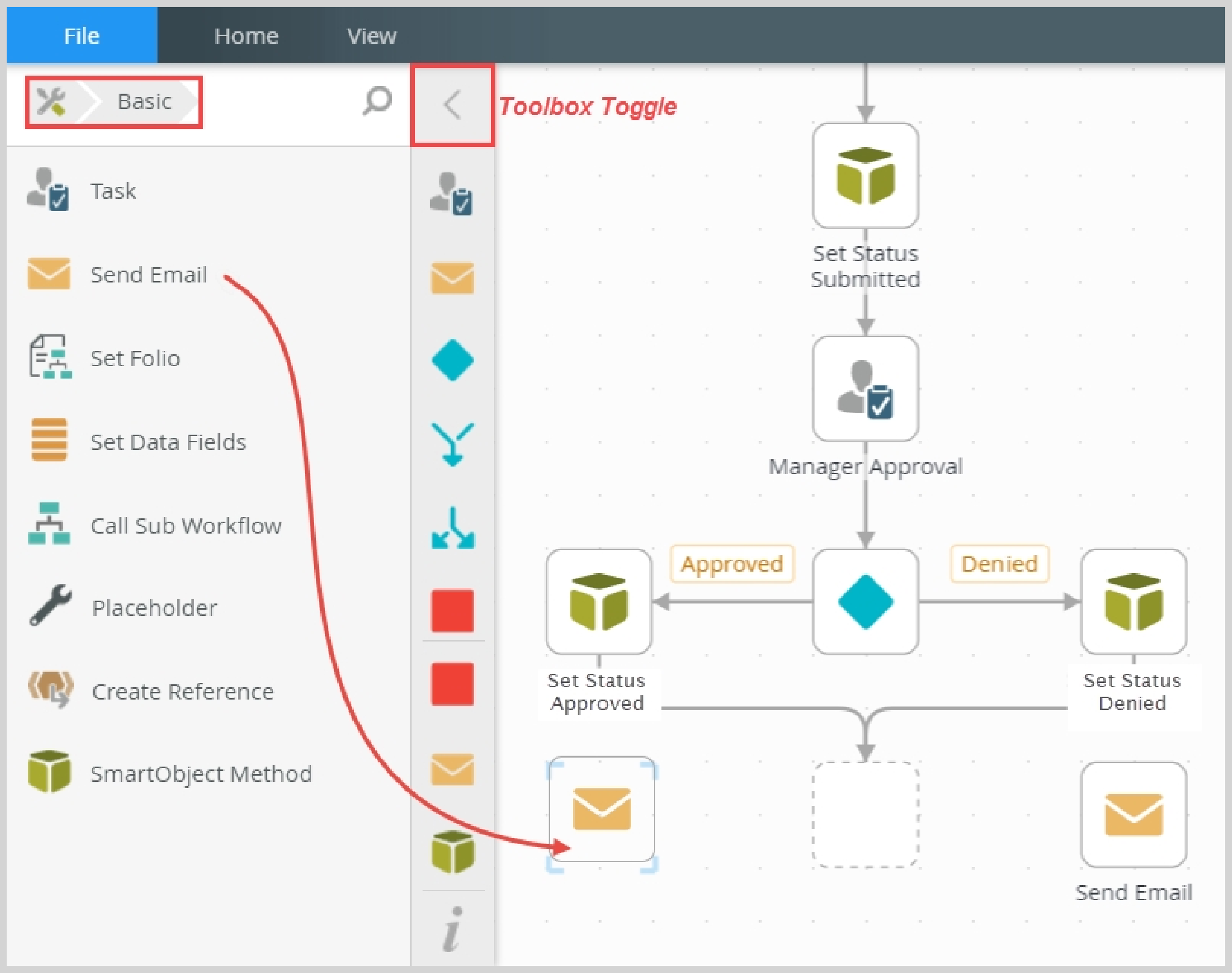
Your Leave Request is Approved: Leave Request S... Leave Request Title

Your manager approved your leave request. Your request details are as follows:

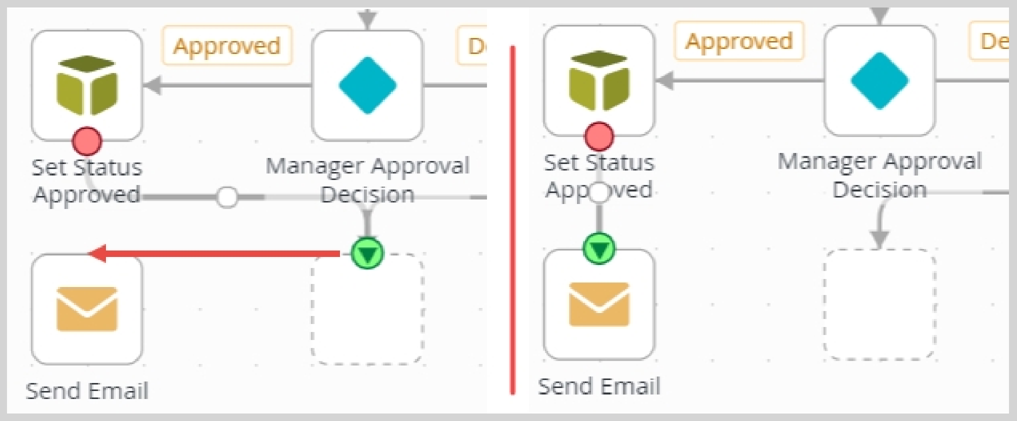
Leave Start Date: Leave Request S... Leave Start Date
Leave End Date: Leave Request S... Leave End Date
Leave Type: Leave Request S... Leave Type

   - Change the Send Email step name to *Email Originator Approved*

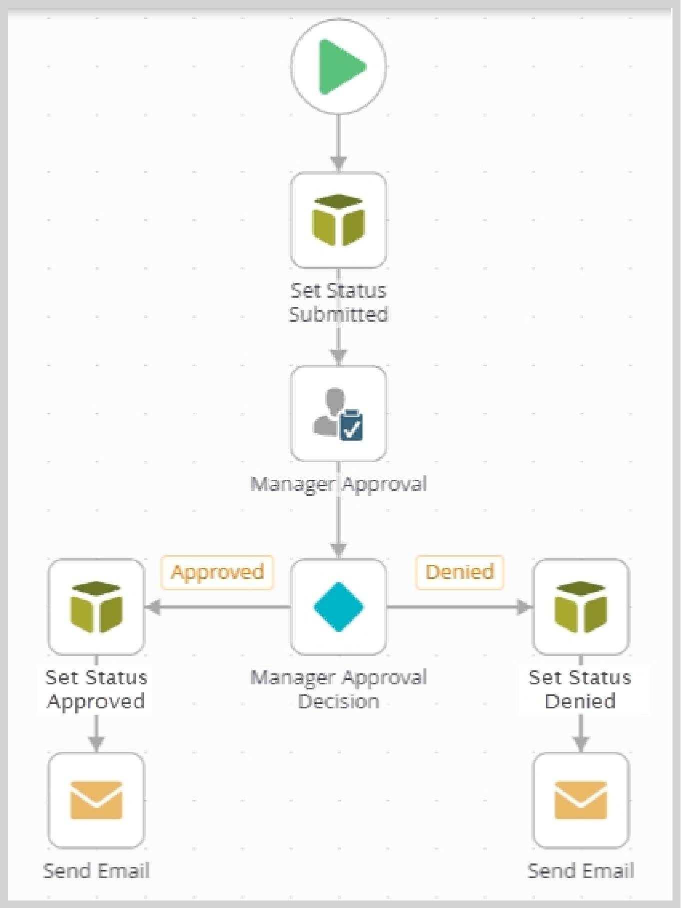
   a. Next you will add and configure the email step for the approved outcome. You will send the email to the workflow originator and let them know the decision from their manager. You will use variables from the Context Browser in your message body. At runtime, K2 replaces the variables with "live" data, which personalizes the email message for the leave requester.
   Expand the **Toolbox** > **Basic** nodes. Drag a **Send Email** step just below the **Set Status Approved** step. Add a **Send Email** step just below the **Set Status Denied** step. You can add as many steps as you need to map out your workflow, then connect them later. You do not have to connect a step before adding another step.



b. Click the **Set Status Approved** outcome line to expose its handles. Drag the handle from the empty step (green triangle) to the **Send Email** step. Repeat this to connect the **Set Status Denied** step to its corresponding **Send Email** step.

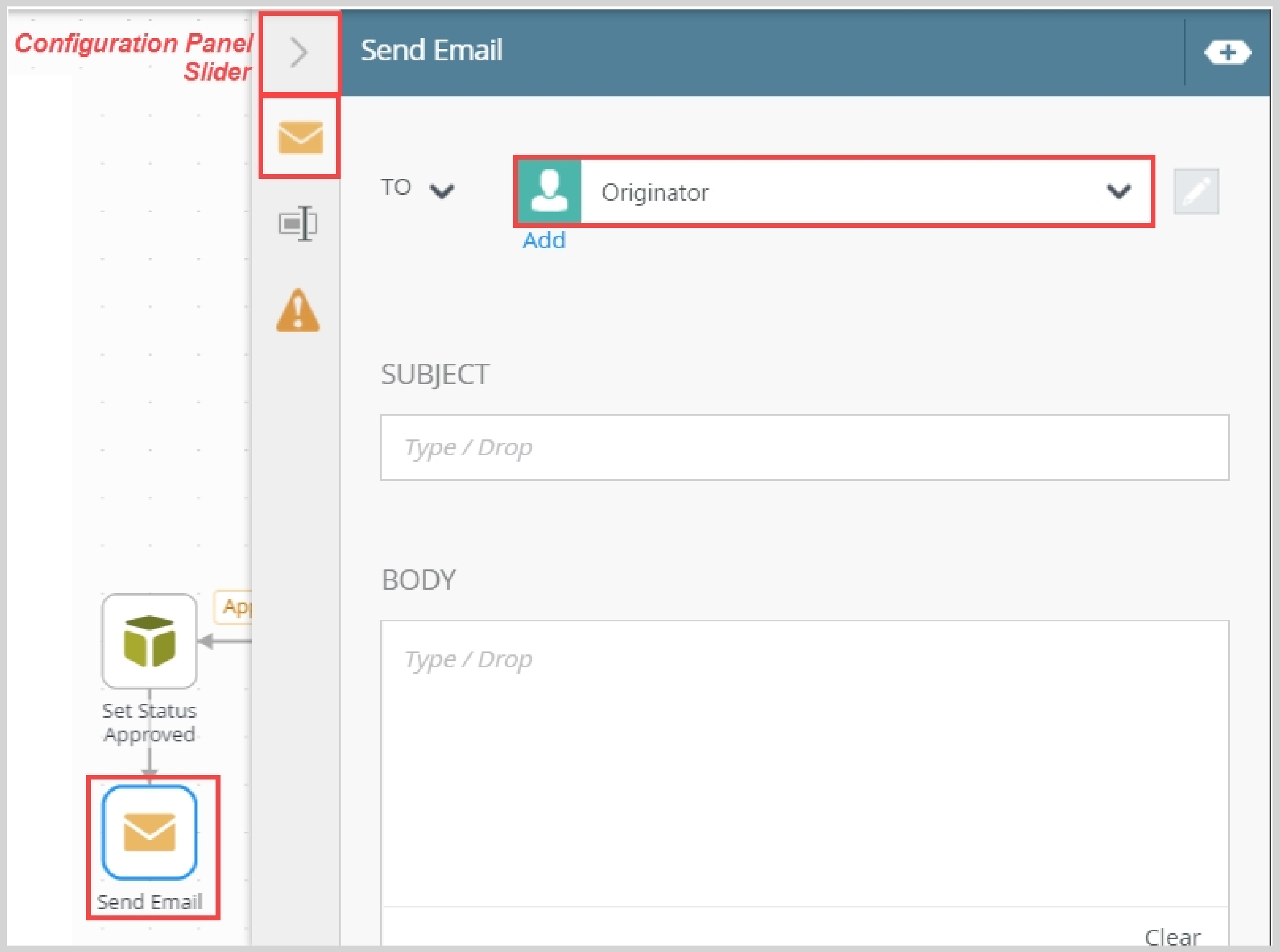


c. The empty placeholder will still remain. Click to highlight it, then click <DELETE> from your keyboard. Your workflow should look like the image below at this point.



d. Next, you will edit both the Send Email steps and configure them for the appropriate outcome (approved and denied). You will use properties from the Context Browser to add dynamic content to your email subject line and message body.
Click to highlight the **Send Email** step for the approved outcome. Expand the **Configuration Panel**. Click the **Notification** tab (if it isn't already active). Confirm the **TO** value is set to the default, **Ori-**

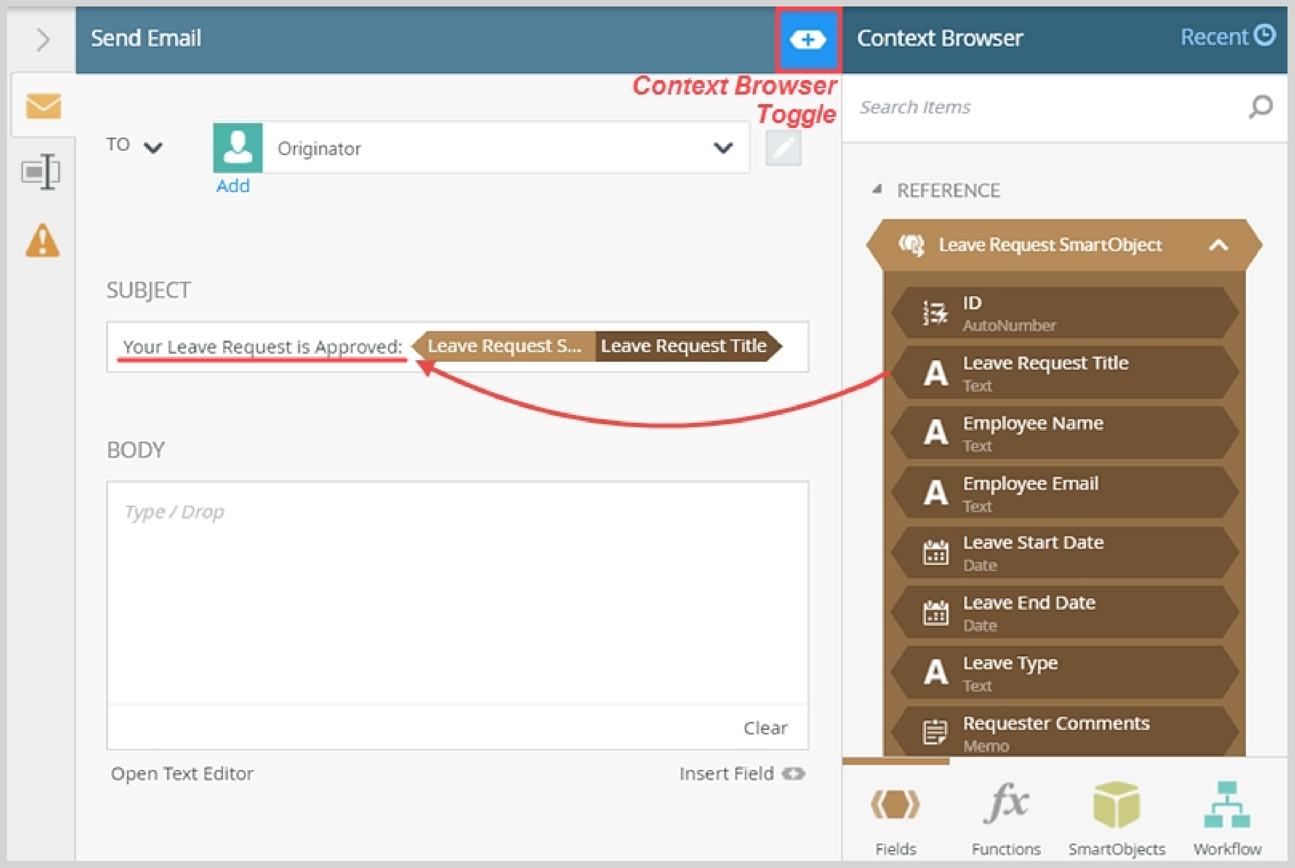
ginator.



e. Expand the **Context Browser**, then expand the **Leave Request SmartObject** under REFERENCES. For the SUBJECT line, enter

Your Leave Request is Approved:

then drag the **Title** property from the reference and append it to the subject line. Once again, at runtime, K2 replaces this variable with "live" content. In this case, the Title text box entry from the leave requester.



Note
The **Context Browser** provides access to data about the current user and process, as well as functions and SmartObject data. You can use variables dragged from the Context Browser throughout the configuration of workflow steps. At runtime, K2 replaces the variables with "live" data. There are four tabs:

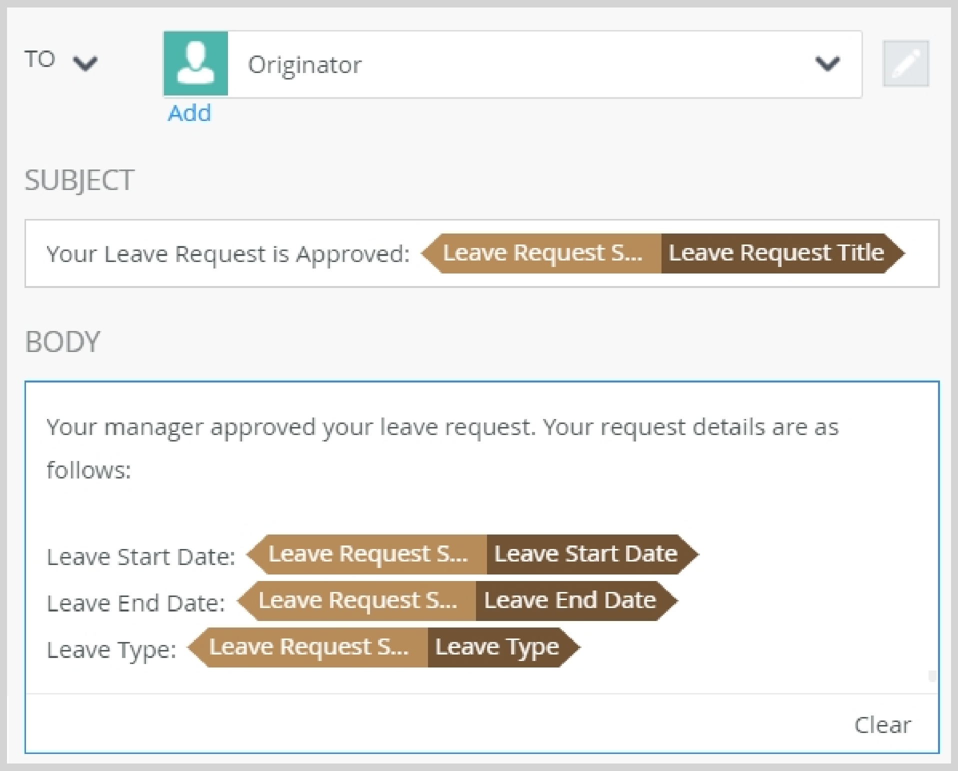
- **Fields**: This section contains references to SharePoint lists, SmartForms, and SmartObjects. You can also create variables to hold and use data within the workflow itself, such as to store record IDs.
- **Functions**: This section contains tools for transforming data. Some examples include performing calculations, formatting text, and working with date and time values.
- **SmartObjects**: This section contains system and custom SmartObject references.
- **Workflow**: This section contains process variables. Examples include the originator's name, email, manager, and folio.

f. Using the same approach as above, enter the following into the BODY. Replace the bracketed text with the appropriate **Leave Request SmartObject** references.

Your manager approved your leave request. Your request details are as follows:

***Leave Start Date**:* [Leave Start Date]
***Leave End Date**:* [Leave End Date]

Leave Type: [Leave Type]



TO
Originator
Add

SUBJECT

Your Leave Request is Approved: Leave Request S... Leave Request Title

BODY

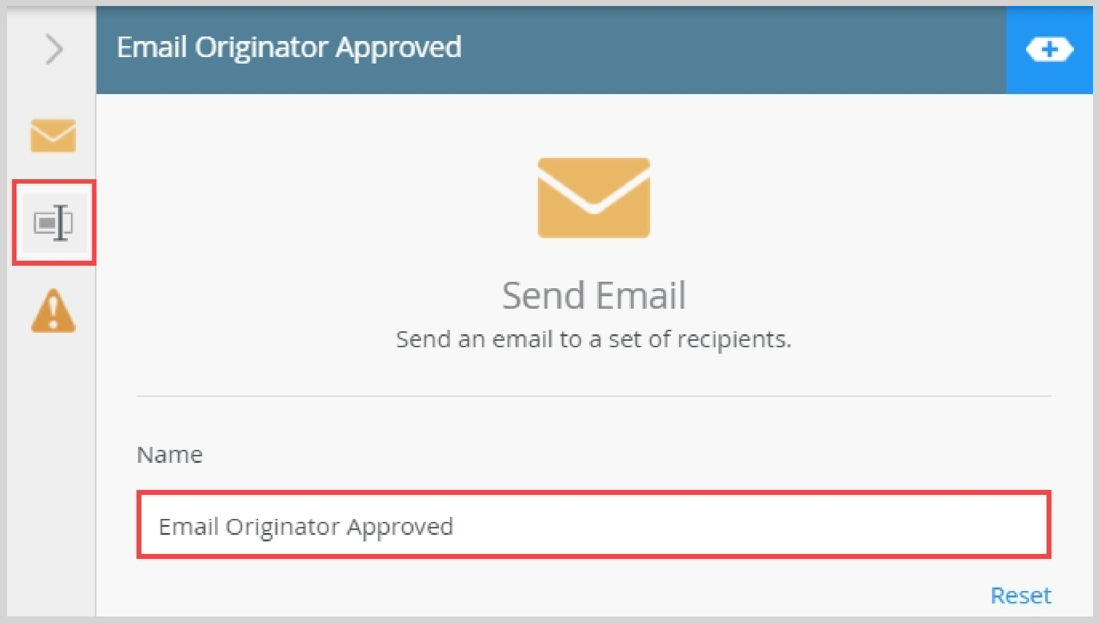
Your manager approved your leave request. Your request details are as follows:

Leave Start Date: Leave Request S... Leave Start Date
Leave End Date: Leave Request S... Leave End Date
Leave Type: Leave Request S... Leave Type

g. Click the **General Properties** tab. Change the step **Name** to

Email Originator Approved

then collapse the **Context Browser** and the **Configuration Panel**. Renaming a step makes it easier for you to tell at a glance what the step does.

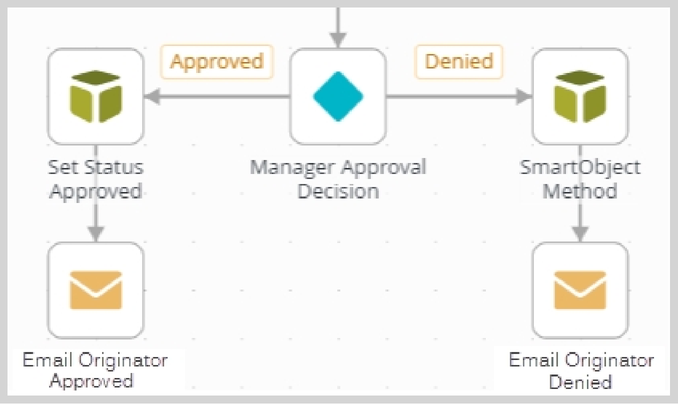


Email Originator Approved

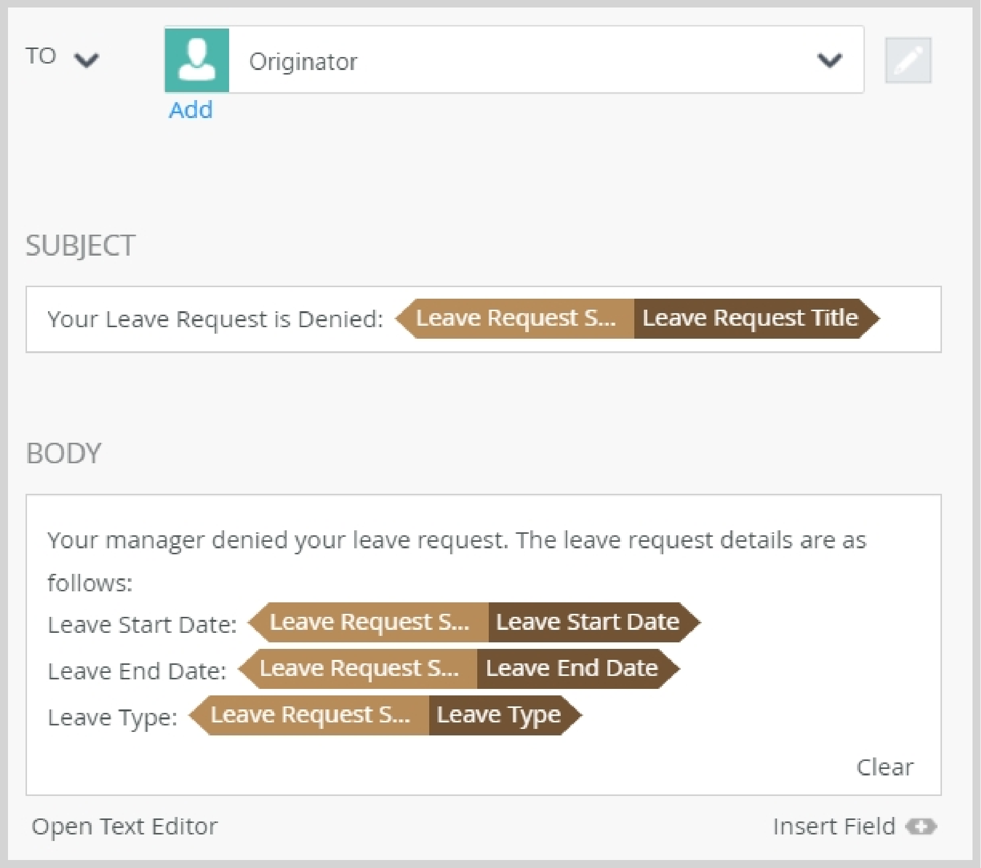
Send Email
Send an email to a set of recipients.

Name
Email Originator Approved

3. Repeat the steps above for the **Denied** outcome. Configure the email recipient as the originator, then customize the SUBJECT line and message BODY using variables from the Leave Request SmartObject reference. Don't forget to rename the step as well.



a. Using the same steps as above, configure the **email** for the **denied** outcome. The recipient should be the originator. Expand the Context Browser and user Leave Request SmartObject references in your SUBJECT line and message BODY. The completed email should look similar to the image below. Be sure to change the step name as well to *Email Originator Denied*.



TO: Originator

SUBJECT

Your Leave Request is Denied: Leave Request S... Leave Request Title

BODY

Your manager denied your leave request. The leave request details are as follows:
Leave Start Date: Leave Request S... Leave Start Date
Leave End Date: Leave Request S... Leave End Date
Leave Type: Leave Request S... Leave Type

Review

In this step, you added Send Email steps and customized the email using variables found in the Context Browser. K2 replaces the variables with "live" values at runtime. Using variables allows you to personalize workflow content. You also learned how to delete a step (the End step) and how to redirect outcome lines to another step.

In the next step, you will add a rework loop, which sends the workflow back to the leave requester to either rework and resubmit the request, or cancel the request entirely.

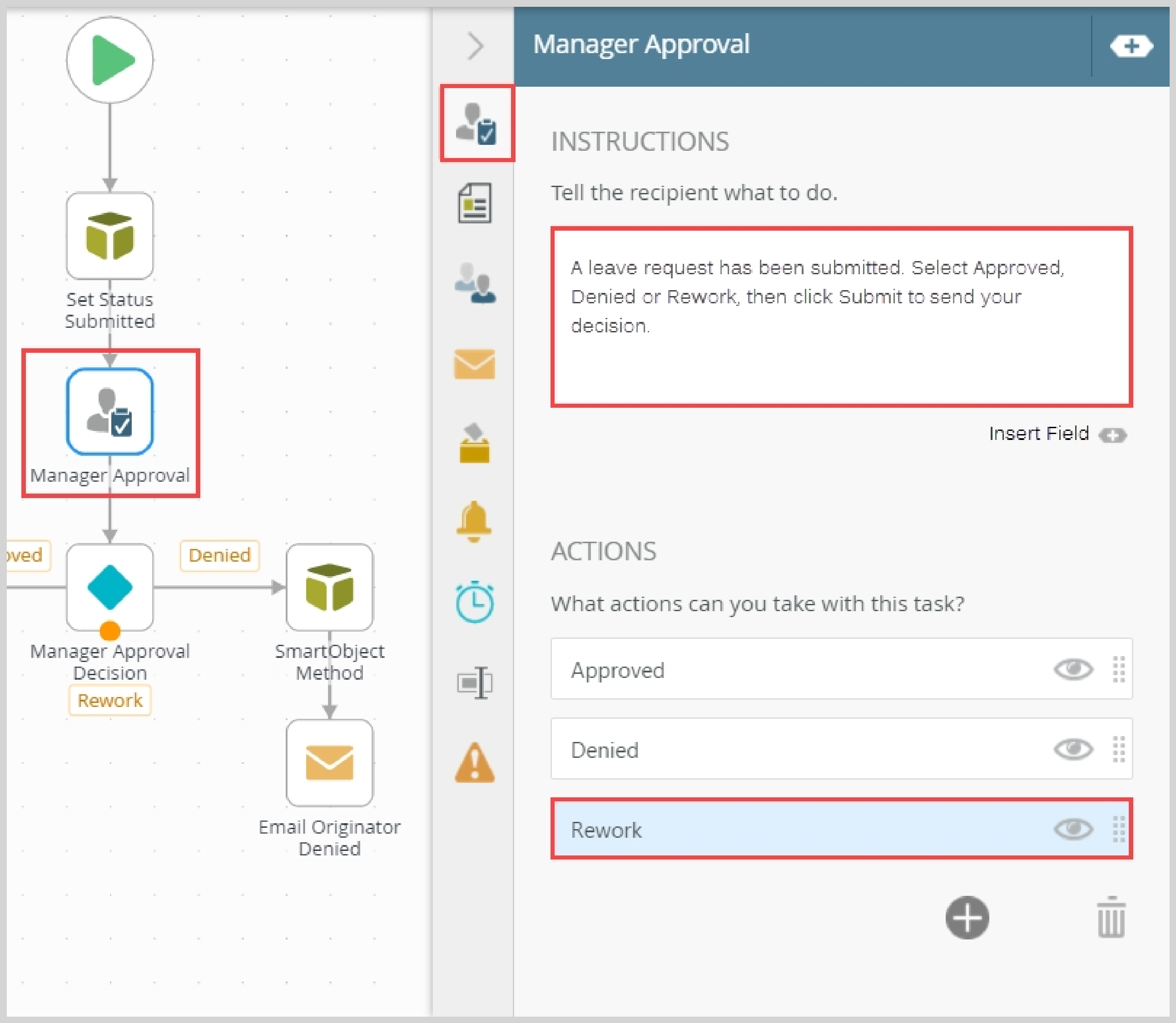
Next Step: 6. Add the Rework Outcome and Requester Rework Task

6. Add the Rework Outcome and Requester Rework Task

In this step, you will add a third action, Rework, to the Manager Approval step. The approving manager can now select either Approved, Denied or Rework. The rework outcome connects to a new task, Requester Rework. The leave requester is the task recipient, who must decide to either resubmit the request or cancel the request. To complete this step, you will add a Decision step and a SmartObject Method step. The Decision step generates the outcomes for the Requester Rework task step, while the SmartObject Method step updates the status property.

1. Begin by adding a third action option for the approving manager so that they can send the request back to the requester for rework.
   Add a third ACTION to the **Manager Approval** task and name it *Rework*. Update the instructions to the manager so that they include the Rework option.
   a. In this step, you will add a third action that allows the approving manager to send the request back for rework. The request routes back to the leave requester, who can either rework and resubmit the request, or cancel the request.
      Select the **Manager Approval** step. Expand the **Configuration Panel**. Click the **Task Detail** tab (if it is not already active). Add a third action called
      Rework
      then update the INSTRUCTIONS so that it includes the Rework action as a decision option. For example,
      A leave request has been submitted. Select Approved, Denied or Rework, then click Submit to send

your decision.



INSTRUCTIONS

Tell the recipient what to do.

A leave request has been submitted. Select Approved, Denied or Rework, then click Submit to send your decision.

ACTIONS

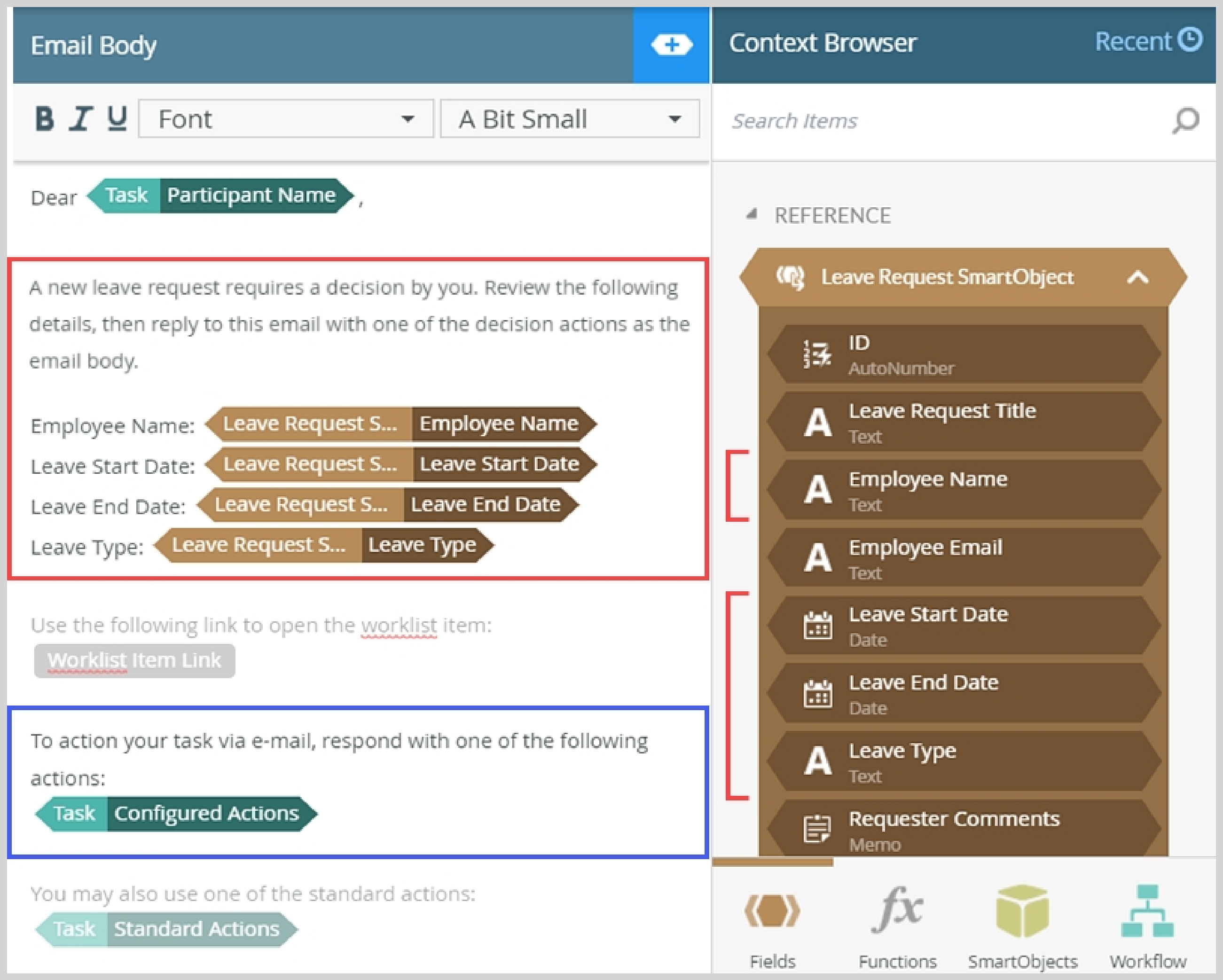
What actions can you take with this task?

Approved

Denied

Rework

2. Customize the task notification email using variables from the Context Browser. At runtime, K2 replaces the variables with live data, which personalized the email message. From the **Notification** tab, add additional content to the SUBJECT line and message BODY using the image below as a guide.



Dear Task Participant Name ,

A new leave request requires a decision by you. Review the following details, then reply to this email with one of the decision actions as the email body.

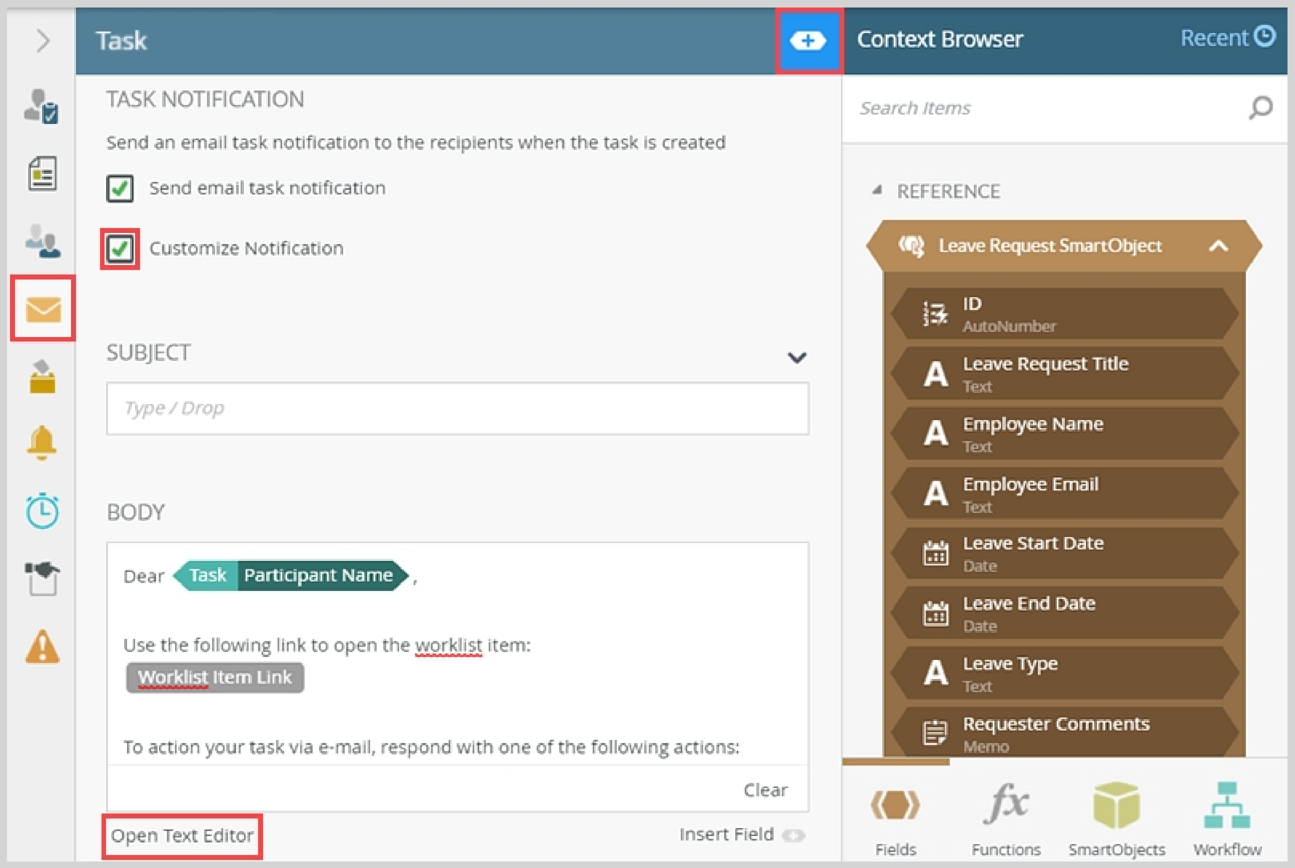
Employee Name: Leave Request S... Employee Name
Leave Start Date: Leave Request S... Leave Start Date
Leave End Date: Leave Request S... Leave End Date
Leave Type: Leave Request S... Leave Type

Use the following link to open the worklist item:
Worklist Item Link

To action your task via e-mail, respond with one of the following actions:
Task Configured Actions

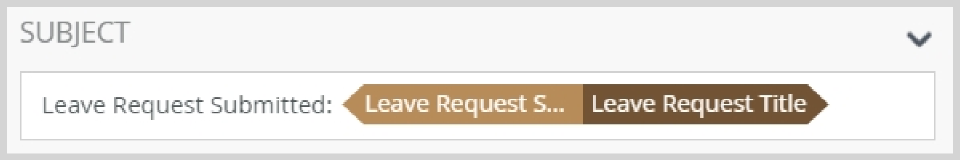
You may also use one of the standard actions:
Task Standard Actions

a. In the next section, you will customize the task notification email. You will update the subject line and edit the message body so that they include variables from the Context Browser. This personalizes the message and provides more in-depth content so that the manager can make a decision directly from the email itself without opening the form.
Click the **Notification** tab. CHECK the box to **Customize Notification**. Expand the **Context Browser**.
Click the link: **Open Text Editor** (located near the bottom of the screen).



The text editor opens in a new pane. The text editor gives you a better view of the email content, making it easier to edit. K2 loads the default task notification content. You are going to add additional text to the email so that the manager can approve, reject or rework the request using SmartActions. Notice the paragraph in the email that contains the properties called **Configured Actions**. (Refer to the image in Step (c), blue outline.) At runtime, the email will display the Manager Approval actions you configured (Approved, Denied, Rework) for this task . Using SmartActions, you can reply to the email with one of the actions as the message body.

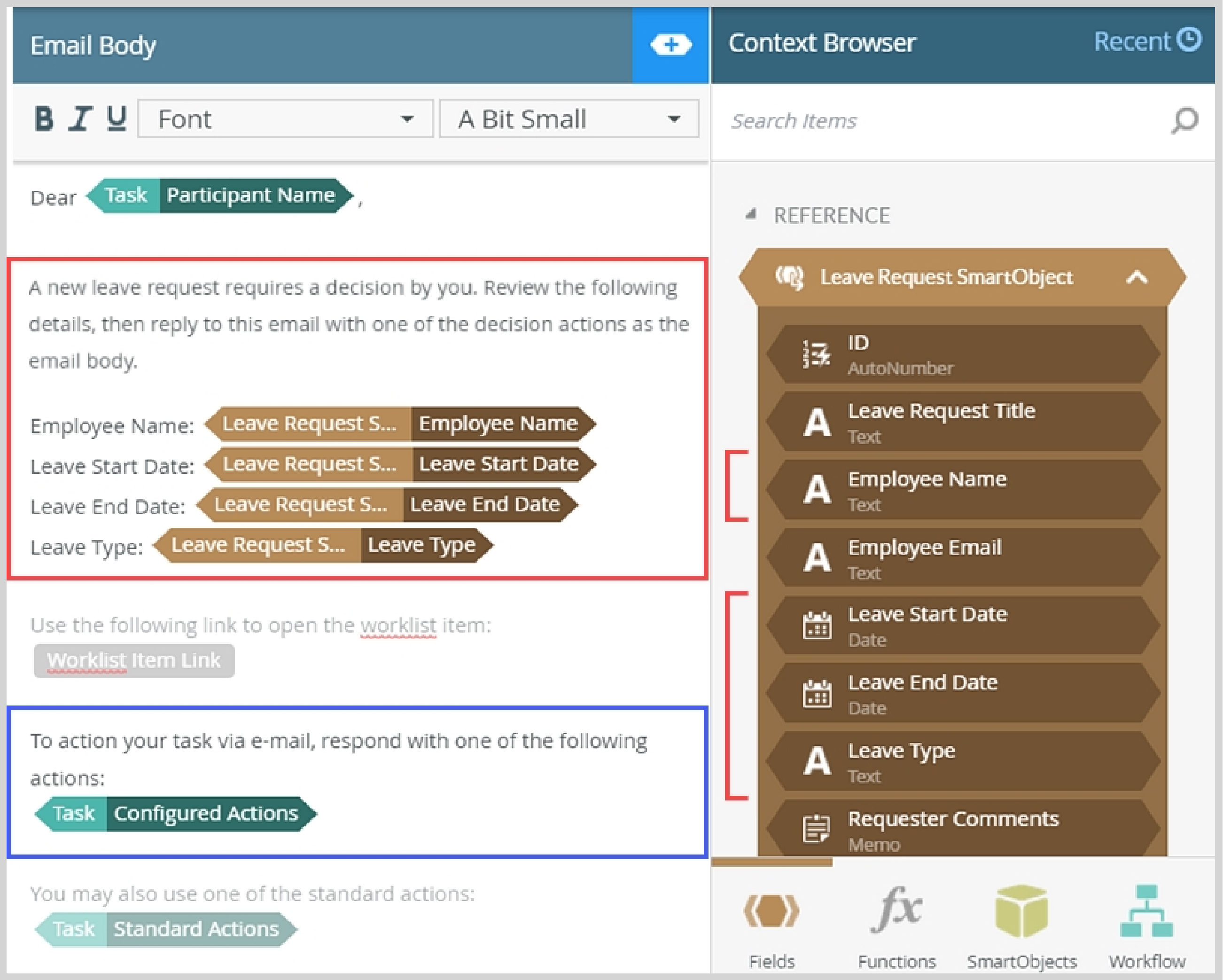
b. For the **Subject** line, enter
Leave Request Submitted: [Leave Request Title]
then append the **Leave Request Title** to the end of the subject line, replacing the bracketed text. The title property is located in the **Leave Request SmartObject** under REFERENCE. At runtime, K2 replaces the variables with "live" content. For this application, the subject line includes the Leave Request Title text box entered by the leave requester.



c. In the email body, just below the *Dear Participant* line, enter the following text, then drag variables from the Leave Request SmartObject reference in place of the bracketed text.
A new leave request requires a decision by you. Review the following details, then reply to this email with one of the decision actions as the email body.

Employee Name: [Employee Name]

Leave Start Date: [Leave Start Date]
Leave End Date: [Leave End Date]
Leave Type: [Leave Type]



Dear Task Participant Name ,

A new leave request requires a decision by you. Review the following details, then reply to this email with one of the decision actions as the email body.

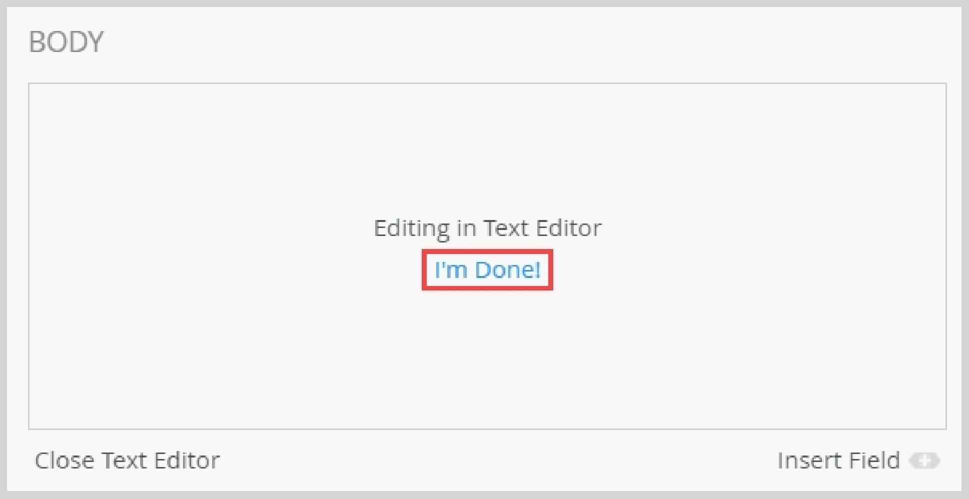
Employee Name: Leave Request S... Employee Name
Leave Start Date: Leave Request S... Leave Start Date
Leave End Date: Leave Request S... Leave End Date
Leave Type: Leave Request S... Leave Type

Use the following link to open the worklist item:
Worklist Item Link

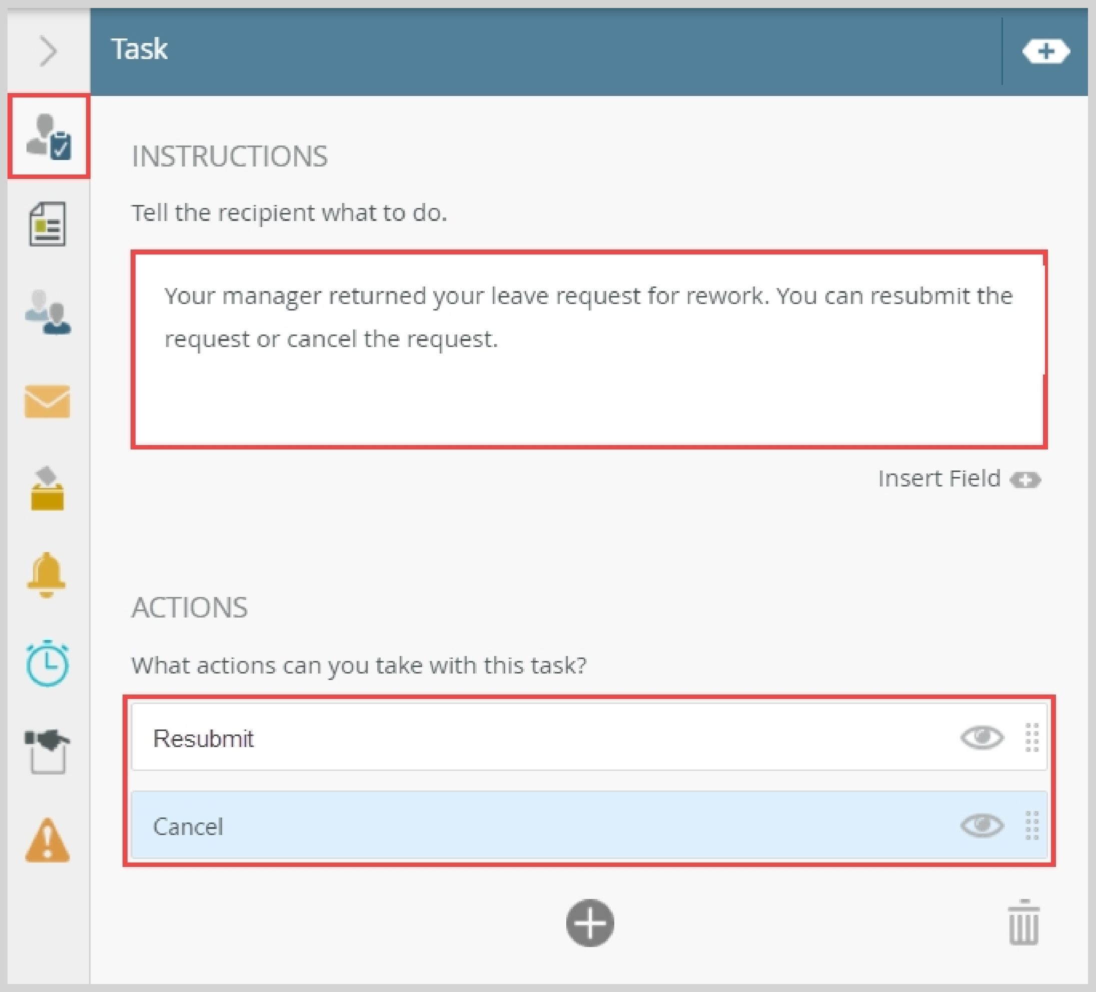
To action your task via e-mail, respond with one of the following actions:
Task Configured Actions

You may also use one of the standard actions:
Task Standard Actions

d. There are no other configurations, so close the **email text editor**. Collapse the **Context Browser**, then collapse the **Configuration Panel**. To close the email text editor, click the **I'm Done** link in the message BODY pane. Collapse the **Configuration Panel**.



3. Next, you will add and configure the rework **Task** step and assign it to the leave requester.
   Add a **Task** step to the design canvas and connect it to the **Rework** outcome. Configure the Task step as follows:
   - For the INSTRUCTIONS, enter *Your manager returned your leave request for rework. You can resubmit the request or cancel the request.*
   - For the ACTIONS, enter two actions: *Resubmit* and *Cancel*.



Task

INSTRUCTIONS

Tell the recipient what to do.

Your manager returned your leave request for rework. You can resubmit the request or cancel the request.

Insert Field

ACTIONS

What actions can you take with this task?

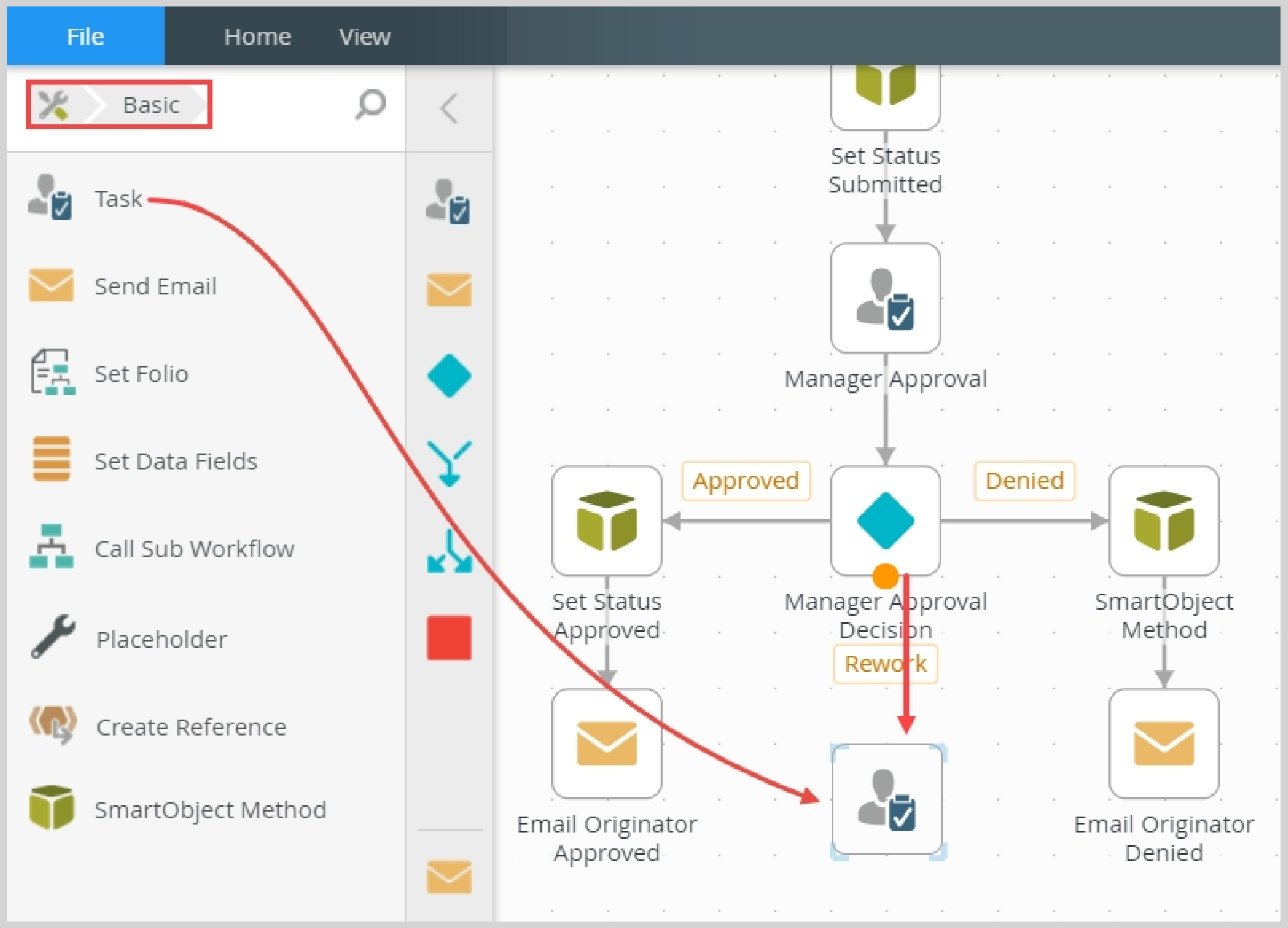
Resubmit

Cancel

   - For the FORM TYPE, select **SmartForm**. For the FORM, select the **Leave Request Form**. The form configuration wizard launches.
     - For the REFERENCES, keep the default settings.
     - For the OPEN TASK, select **Help me configure it**.
     - For the FORM STATE, create a new state called **Workflow Task** (the default state name).
     - Keep the remaining default values for the form wizard.
   - For the task RECIPIENT, keep the default **Originator**.
   - For the TASK NOTIFICATION, you will not customize this email. Keep the default settings as is.
   - For the GENERAL PROPERTIES, rename the step *Requester Rework*.

   a. Now that you have a third action, you need to configure the steps for the workflow if it follows that outcome. In this case, you will add a new Task step and assign it to the leave requester. The requester must either rework and resubmit the request, or cancel it entirely.
   From the **Toolbox** > **Basic** node, drag a **Task** step just below the **Decision** step. Connect the **Rework**

outcome to the **Task** step.



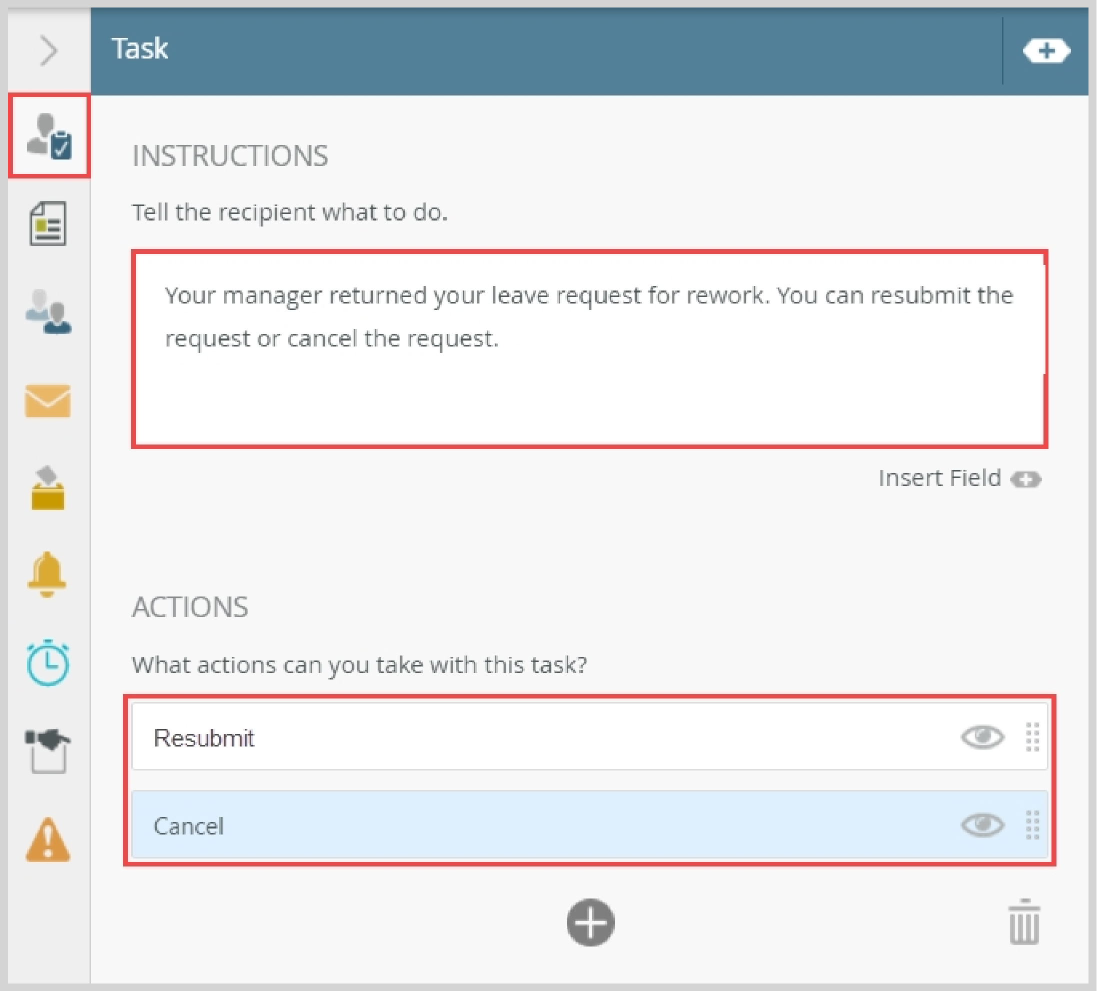
b. Next, you will configure the task step for the leave requester. Select the **Task** step, then expand the **Configuration Panel**. Click the **Task Detail** tab (if is not already active). For the INSTRUCTIONS, enter

Your manager returned your leave request for rework. You can resubmit the request or cancel the request.

then add the two ACTIONS for the rework task:

Resubmit

Cancel



Task

INSTRUCTIONS

Tell the recipient what to do.

Your manager returned your leave request for rework. You can resubmit the request or cancel the request.

ACTIONS

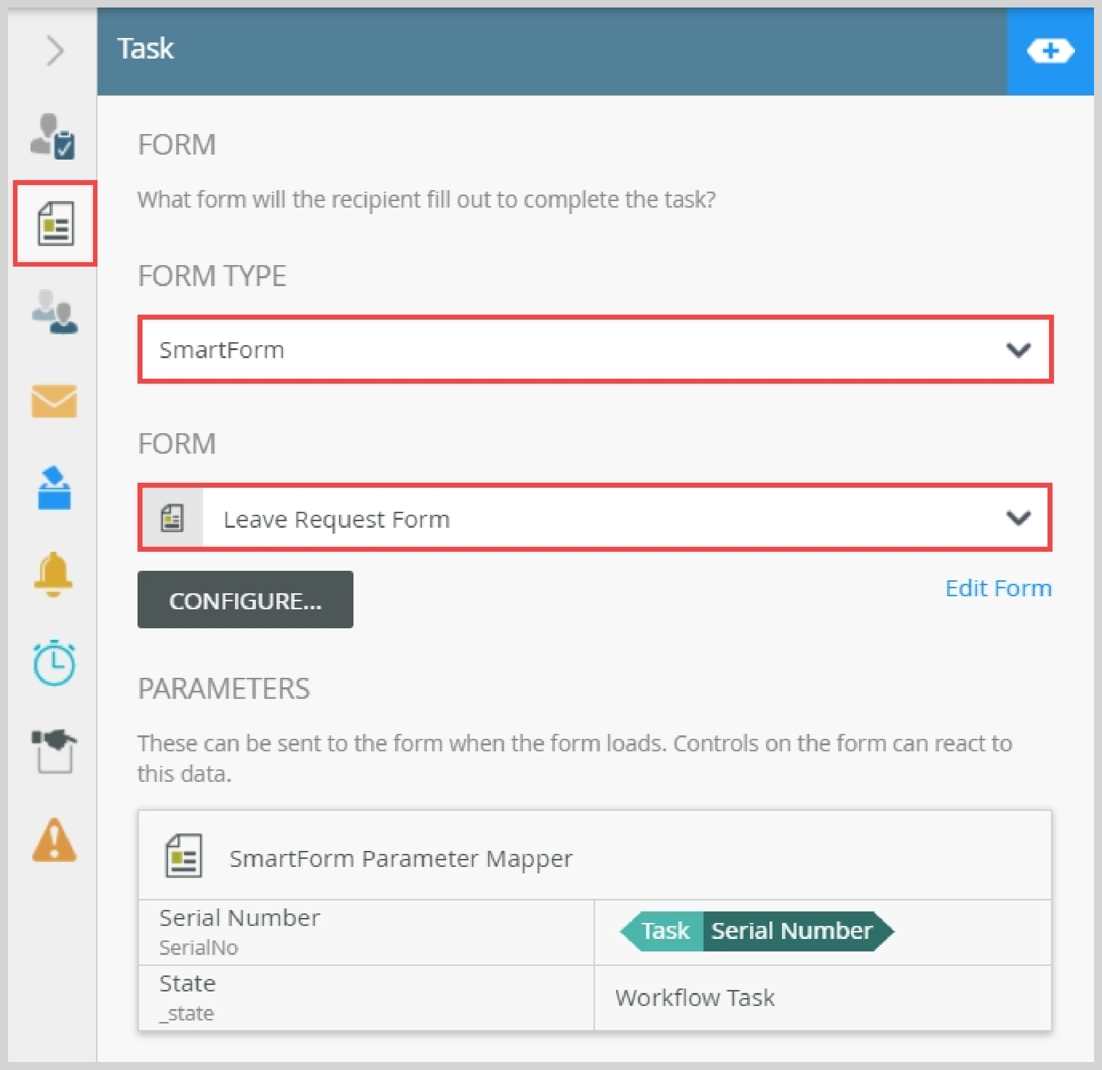
What actions can you take with this task?

Resubmit

Cancel

c. Now you need to add the form for the task. Even though the requester may not actually use the form to respond to the task (more on that later), K2 still needs the form to apply rules for the step and outcome.
Click the **Forms** tab. For the FORM TYPE, select **SmartForm**. For the FORM, navigate to, then select

the **Leave Request Form**.



Task

FORM

What form will the recipient fill out to complete the task?

FORM TYPE

SmartForm

FORM

Leave Request Form

PARAMETERS

These can be sent to the form when the form loads. Controls on the form can react to this data.

SmartForm Parameter Mapper

Serial Number SerialNo	Task Serial Number
State _state	Workflow Task

The Open Task on a Form wizard launches.

d. On the REFERENCES screen, K2 should have found at least one SmartObject, possibly two (Leave Request SmartObject and Leave Types). This lets you know the K2 connection between SmartObject > Form > Workflow is working! Click **NEXT**.

e. On the OPEN TASK screen, select **Help me configure it**, then click **NEXT**.

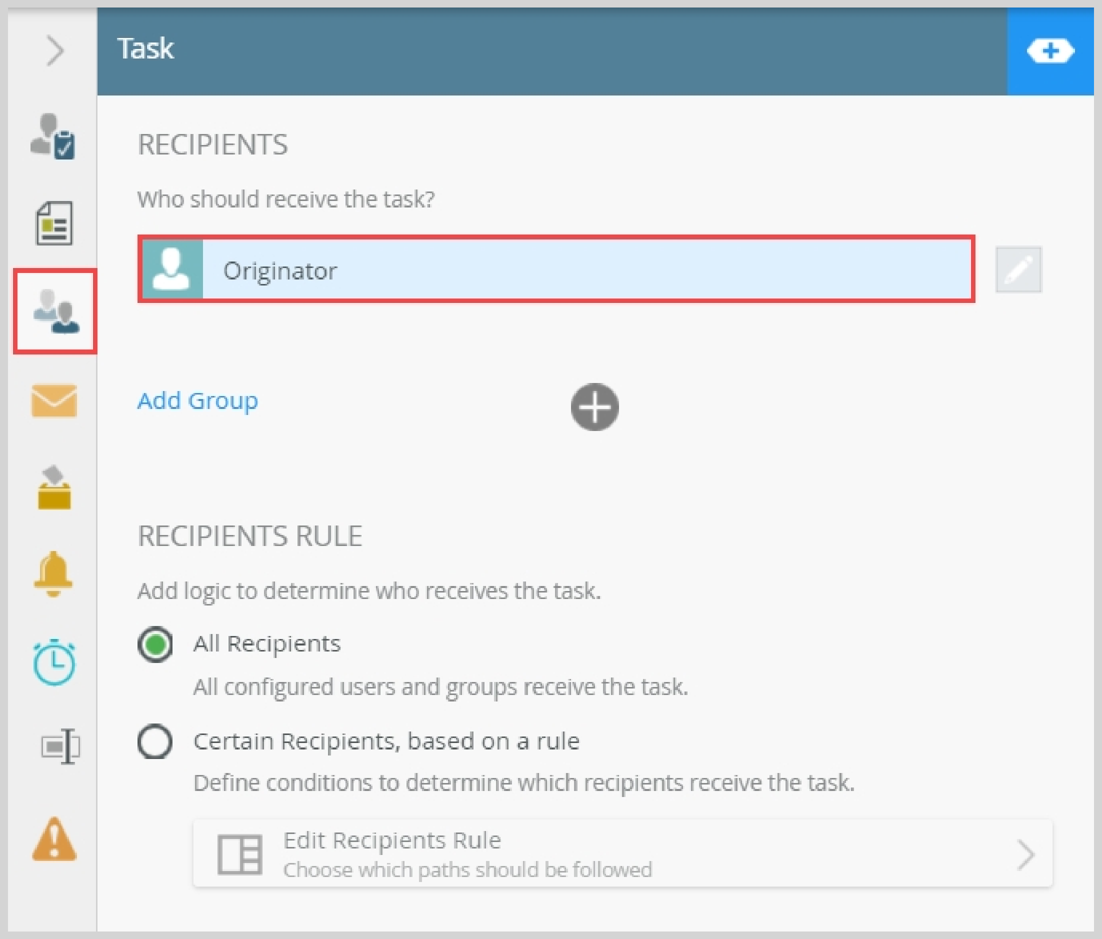
f. On the FORM STATE screen, you can keep the default, **Create a new State**, with **Workflow Task** the default state name. Click **NEXT**.

g. There are no changes on the OPEN TASK RULE ACTION screen, so click **NEXT**.

h. There are no changes on the AFTER SUBMIT ACTION screen, so click **NEXT**.

i. On the summary screen, review the changes that K2 will make, then click **SAVE AND FINISH**. When you see the success check, click **OK**.

j. Next, you will assign the recipient. Click the **Recipients** tab. Since this is the rework task, the recipient should be the leave requester, or workflow originator. Confirm the RECIPIENTS value is **Originator**. You won't make any changes to the recipient rules, so you'll move on to the task notification email settings.



Task

RECIPIENTS

Who should receive the task?

Originator

Add Group

RECIPIENTS RULE

Add logic to determine who receives the task.

● All Recipients
All configured users and groups receive the task.

○ Certain Recipients, based on a rule
Define conditions to determine which recipients receive the task.

Edit Recipients Rule
Choose which paths should be followed

k. You won't customize this task notification email, so continue to renaming the task step. Click the **General Properties** tab. Rename the step

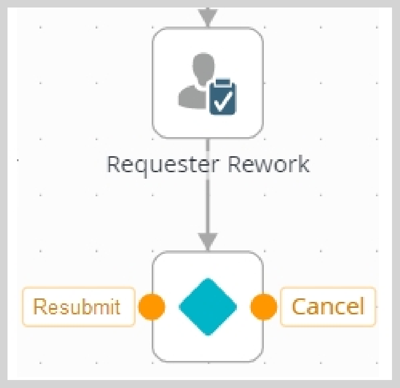
Requester Rework

then collapse the **Configuration Panel**.

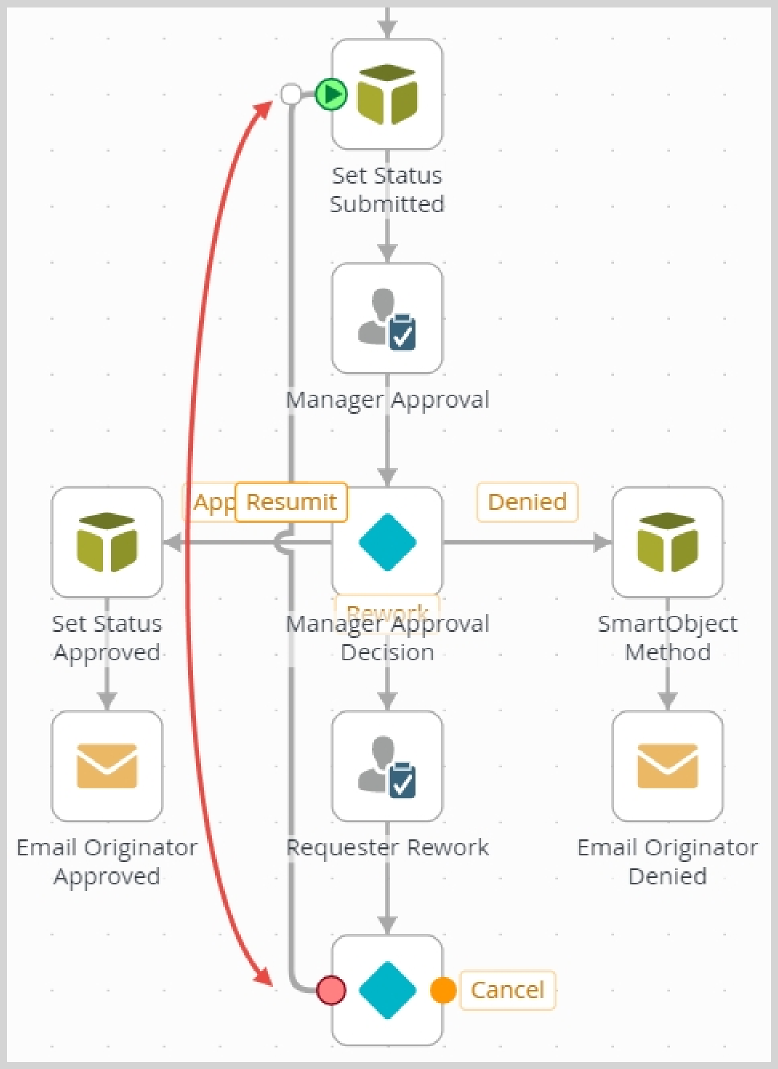
4. Next, you will add a Decision step that generates the outcomes for the Requester Rework task step. Add a **Decision** step and connect the **Requester Rework** step to the Decision step.

   a. From the **Toolbox** > **Logic** node, add a **Decision** step below the **Request Rework** step. Connect the two steps. The Decision step generates the outcomes for the rework step (Resubmit and Cancel). You do not need to edit the Decision step properties, so you can continue to the next step.

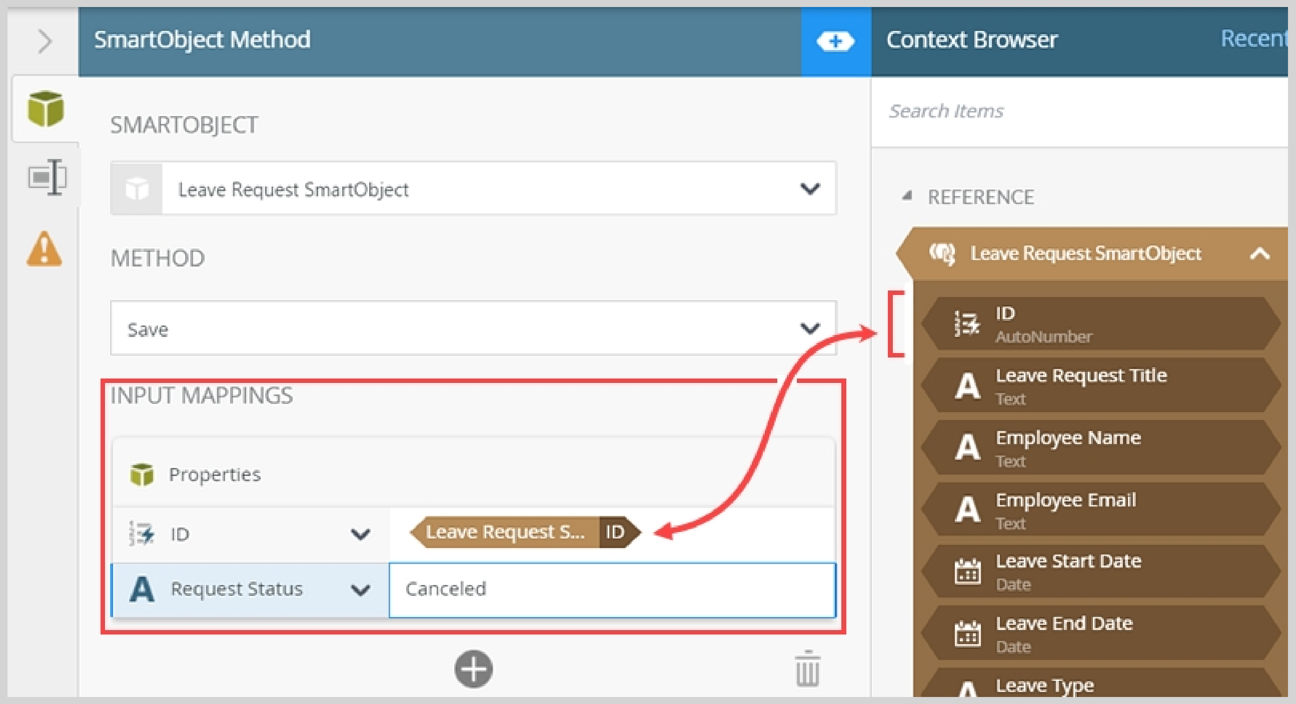
   > **Tip**
   > If you need more canvas space, click anywhere on the design canvas, then drag your mouse to move the workflow around the canvas. For example, you can move the workflow up so that you have more room to add to the bottom steps.



5. Connect the **Resubmit** outcome to the **Set Status Submitted** step.
    a. Next , you will configure the steps for the two rework outcomes (**Resubmit** and **Cancel**). First, you will route the resubmit outcome back to the **Set Status Submitted** step so that the workflow returns to a previous step. For the canceled outcome, you will add a **SmartObject Method** step and set the **Request Status** property to **Canceled**.
    Connect the **Resubmit** outcome to the **Set Status Submitted** step.



6. Now, you need to update the request status property if the leave requester cancels their request. This update appears on the Previous Leave Requests list that both the requester and manager sees when they open the Leave Request Form. Add a new **SmartObject Method** step for the **Cancel** outcome. For the INPUT MAPPINGS, select both the **ID** and **Request Status** properties. Bind the reference **Leave Request SmartObject** > **ID** to the INPUT MAPPINGS **ID**. For the **Request Status** value, enter *Canceled*. **Rename** the SmartObject Method step to *Set Status Canceled*.


SmartObject Method

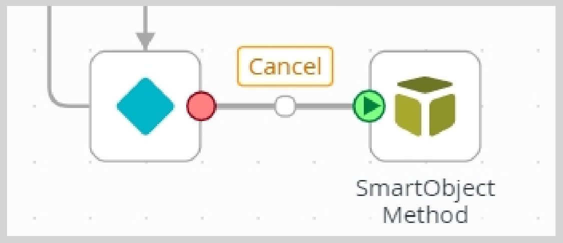
SMARTOBJECT: Leave Request SmartObject

METHOD: Save

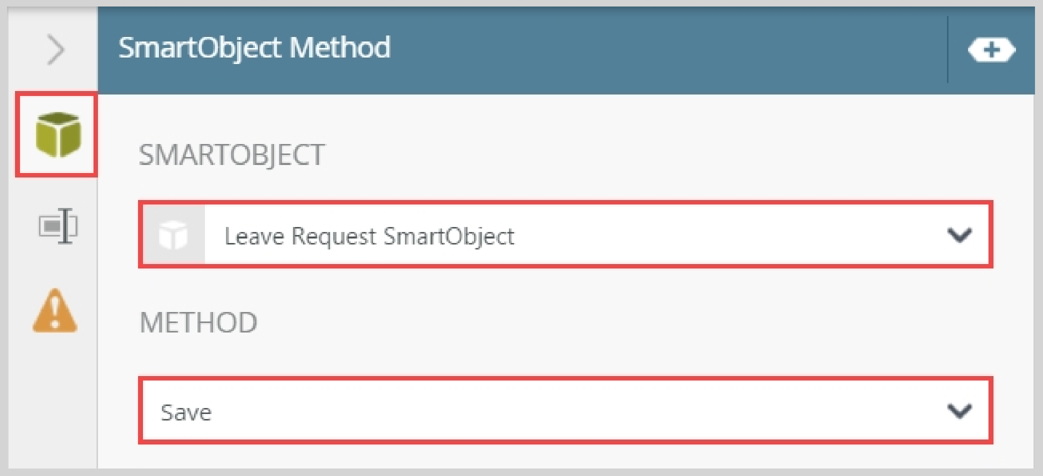
INPUT MAPPINGS

Properties	
ID	Leave Request S... ID
Request Status	Canceled

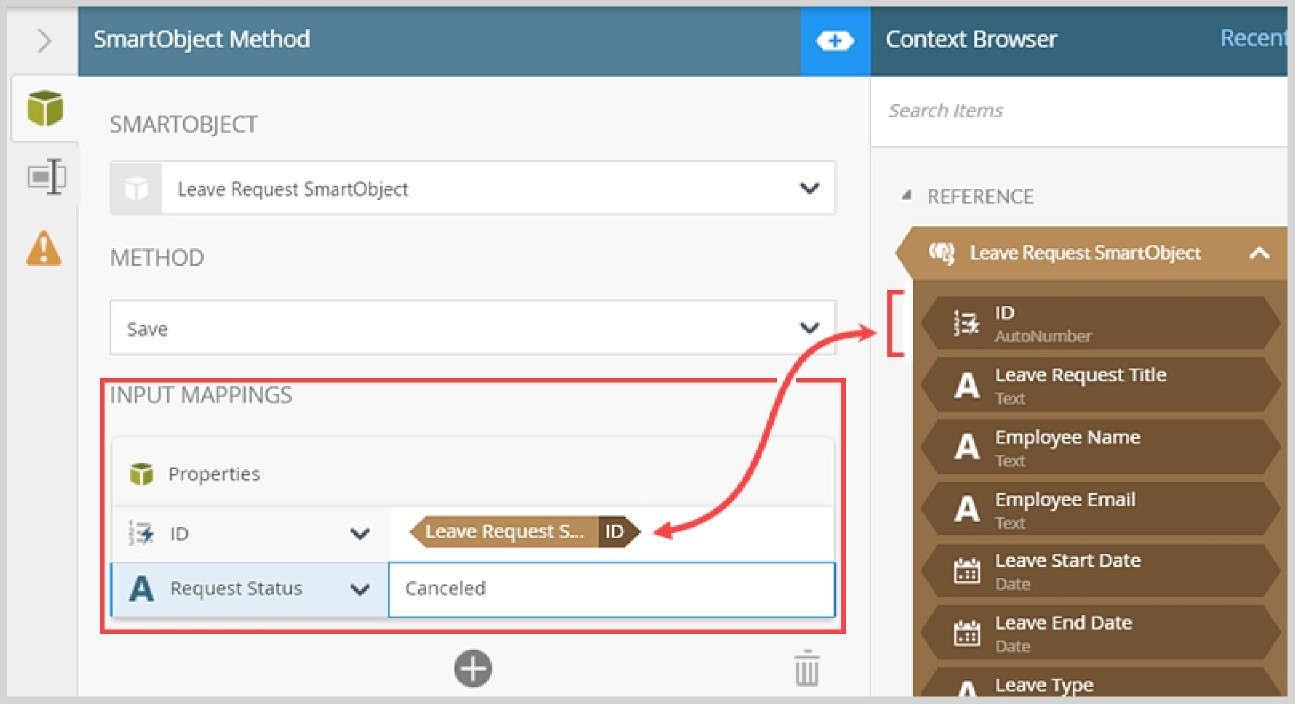
a. From the **Toolbox** > **Basic** node, drag a **SmartObject Method** step so that it aligns with the **Cancel** outcome. **Connect** the Cancel outcome to the **SmartObject Method** step.



b. Select the **SmartObject Method** step. Expand the **Configuration Panel** and click the **SmartObject Detail** tab. Select the **Leave Request SmartObject**. Select the **Save** method. Remember that "save" is the equivalent of "update" in K2 terms.

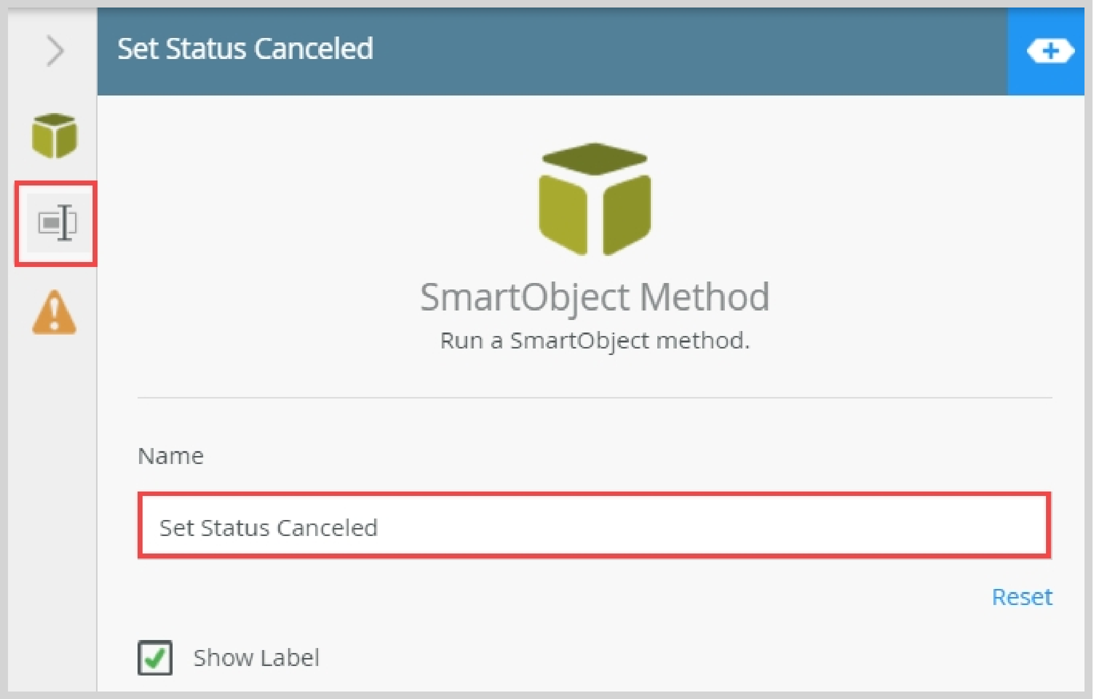


c. Expand the **Context Browser**, then expand the **Leave Request SmartObject** under the REFERENCE heading. In the center pane, click the **Add (+)** icon under INPUT MAPPINGS. Select the **ID** and **Request Status** properties. (You can select both at the same time, then click outside the list menu to close it.) Drag the reference **ID** property next to the **ID** input mapping. For the **Request Status** input mapping, enter
Canceled.
K2 needs to know *which* record to update. So you map the ID of the current record back to the SmartObject > ID. Then, you can update any other properties for this same record as you need to.



d. Once again, you will give the step a friendly name. Click the **General Properties** tab. Change the step **Name** to
Set Status Canceled

then collapse the **Configuration Panel**.



Set Status Canceled

SmartObject Method

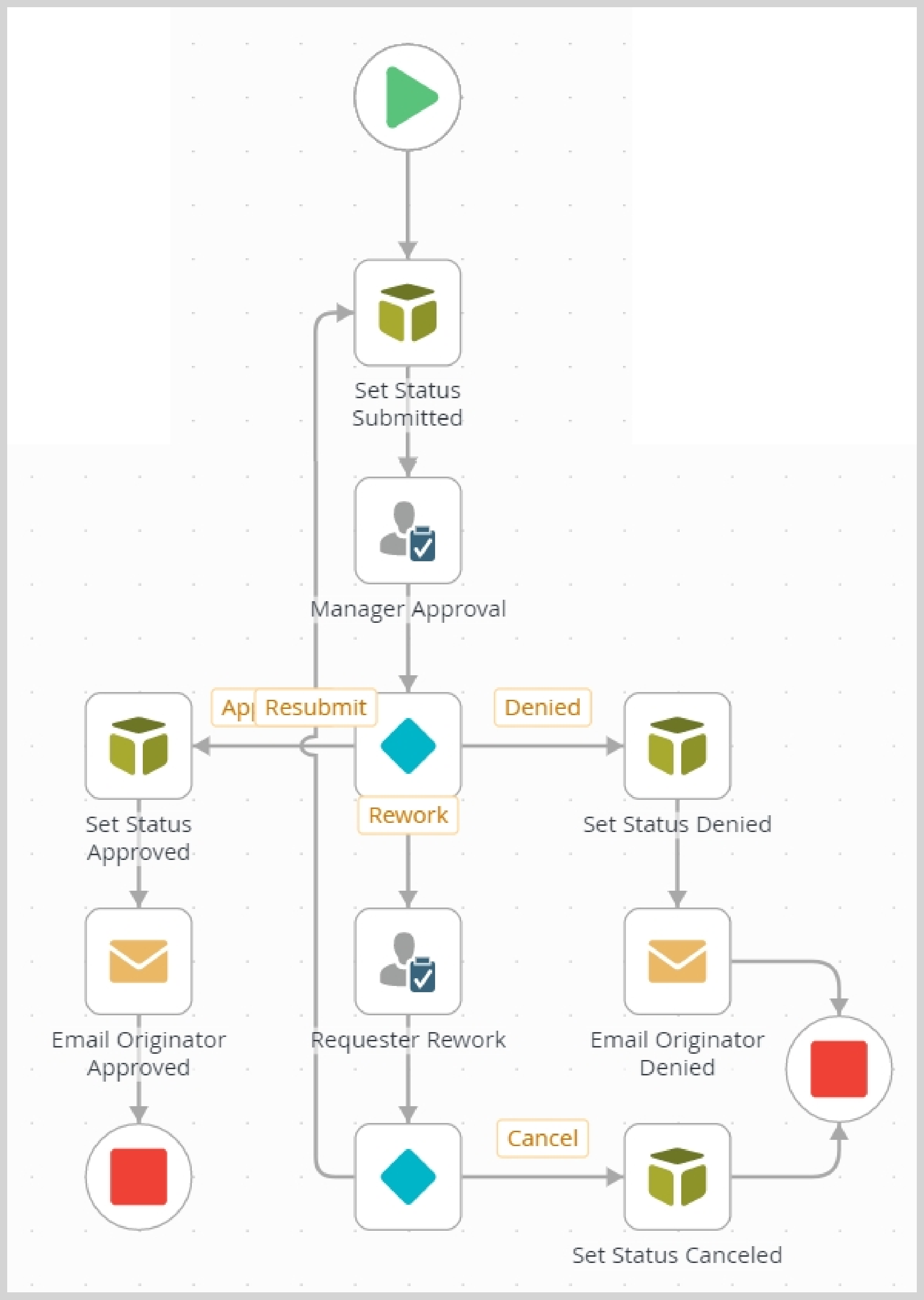
Run a SmartObject method.

Name

Set Status Canceled

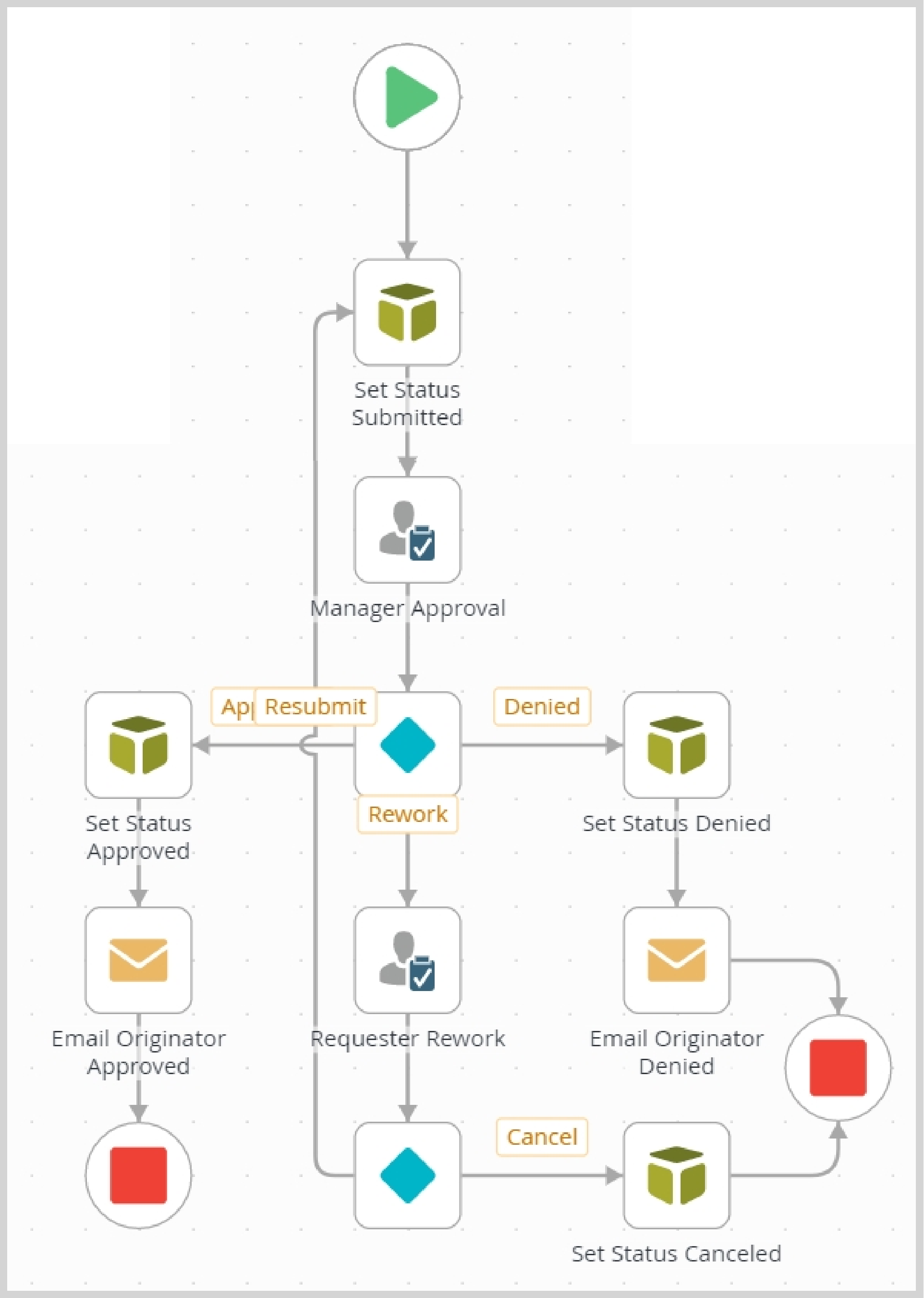
☑ Show Label

7. To complete the extended workflow, add **End** steps and connect them to the **Email Originator Approved**, **Email Originator Denied** and **Set Status Canceled** steps.



a. Next, you will add End steps for the three Manager Approval outcomes (Approved, Denied and Rework). While not required, End steps provide a clear sign that the workflow has come to an end. Other than a visual indicator, the End step has no other function.
From the **Toolbox** > **Logic** node, add an **End** step just below the **Email Originator Approved** step. Connect the two steps. Add another **End** step and align it near the **Email Originator Denied** and **Set Status Canceled** steps. Connect the email step to the End step. Connect the set status step to the same End step. Use the image below as a guide if necessary.

Your completed workflow should look like the image below.



Review

In this step, you added a third action, Rework, for the Manager Approval task. You customized the task notification email to include leave request details. You added a new Task step, Requester Rework, for the Rework outcome. The leave requester is the task recipient and they must decide to either resubmit or cancel the request. You connected the Resubmit outcome to the first status request step and this creates a looping effect. The loop will repeat itself until the manager or the requester chooses a different outcome. Finally, you added End steps. While they are not required, End steps are clear indicators of completed outcomes.

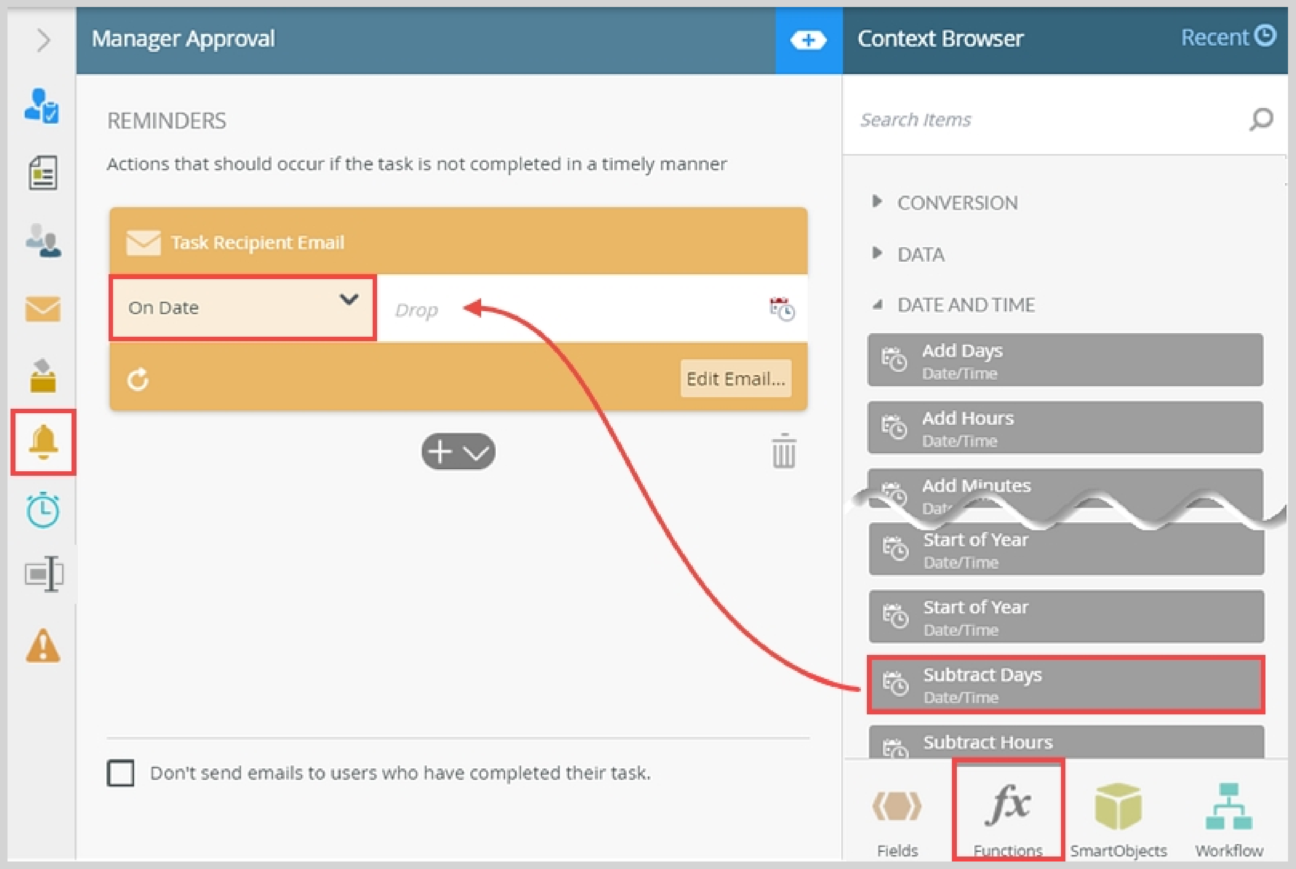
Next Step: 7. Add a Reminder to the Manager Approval Step

7. Add a Reminder to the Manager Approval Step

In this step you will add a reminder to the Manager Approval step. When a recipient has not responded in the time allotted, the reminder action will fire. Reminders can be as simple as an email reminder, or more complex, such as redirecting the task to another user. Reminders can repeat themselves. For example, sending an email reminder every three days. Reminders are an effective tool for keeping workflows moving along and on time.

> **Tip**
> Be careful not to crowd user's in-boxes with reminder notices! Keep in mind the task recipient. For example, should you send multiple email reminders to a manager or department director? Consider assigning the reminder to the workflow originator, who can follow up with the task recipient.

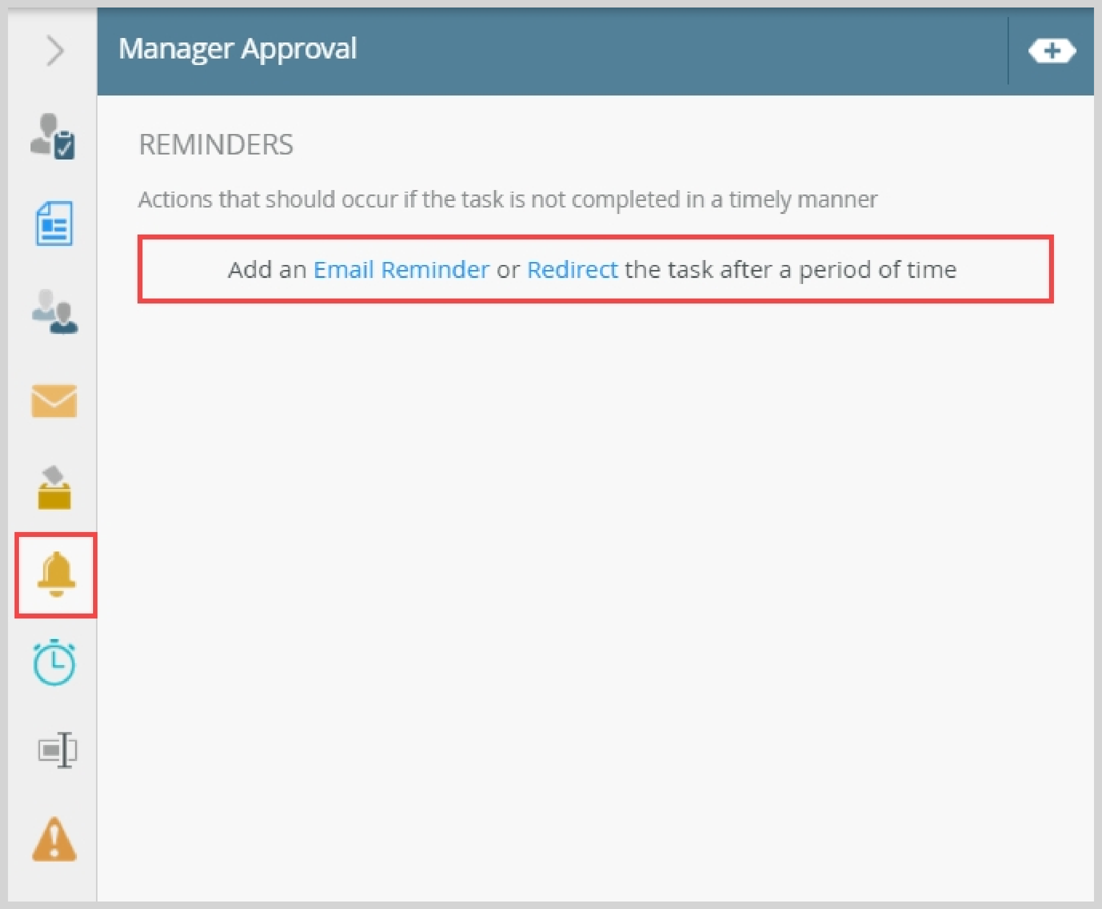
1. In the Configuration Panel, add an **email reminder** for the **Manager Approval** step. Access the settings from the **Reminders** tab. You want the reminder to fire if the manager has not responded to the Manager Approval task **two days** prior to the **Leave Start Date**. You can use the **Subtract Days** function from the **Functions** > **Date and Time** menu in the Context Browser to accomplish this. This reminder should send an email to the workflow originator (leave requester) and the approving manager.



   a. If the approving manager has not responded to the Manager Approval task two days prior to the leave start date, you want to send them an email reminder. You will include the leave requester as well, so that they can follow up with the manager if necessary. You can configure reminders from the

Reminders tab in the Configuration Panel for the Task step.
Select the **Manager Approval** step. Expand the **Configuration Panel**, then click the **Reminders** tab.
Click the link to **Add an Email Reminder**.

> **Note**
> The easiest way to remind a recipient of an unactioned task is to send an email. You can configure the email to send just once, or periodically, such as every other day. A more complex method to handling unactioned tasks is to redirect the task to another user. You can also use a combination of the two. For example, sending three email reminders, then redirecting the task to another user. You may also hear the term "escalation". Reminders and escalations are the same thing.



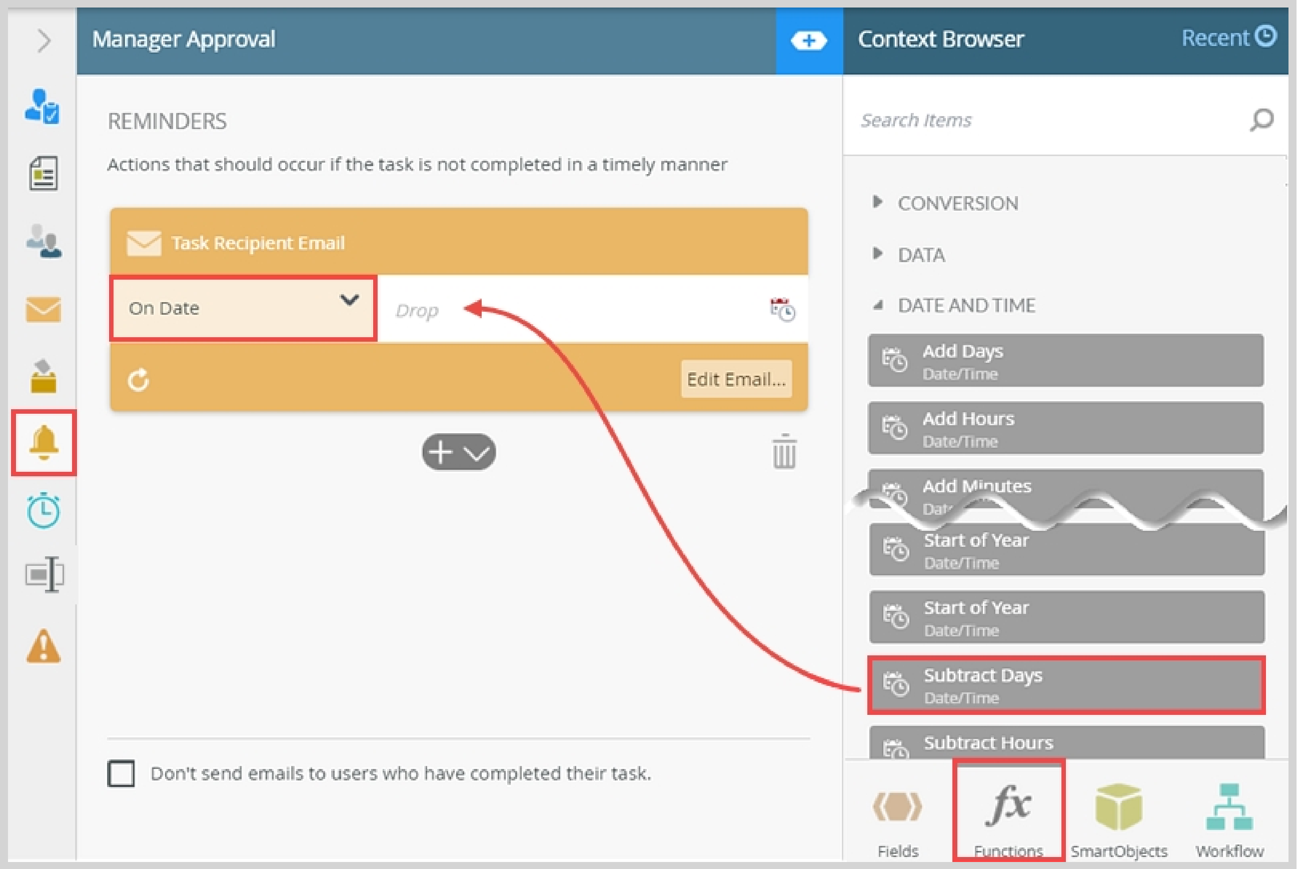
REMINDERS

Actions that should occur if the task is not completed in a timely manner

Add an Email Reminder or Redirect the task after a period of time

b. Change the reminder date to **On Date**. You will configure the reminder to fire two days prior to the Leave Start Date. Expand the **Context Browser**. Click the **Functions** tab. The function you are adding will create a new (reminder) date by taking the Leave Start Date, minus two days. Expand the DATE

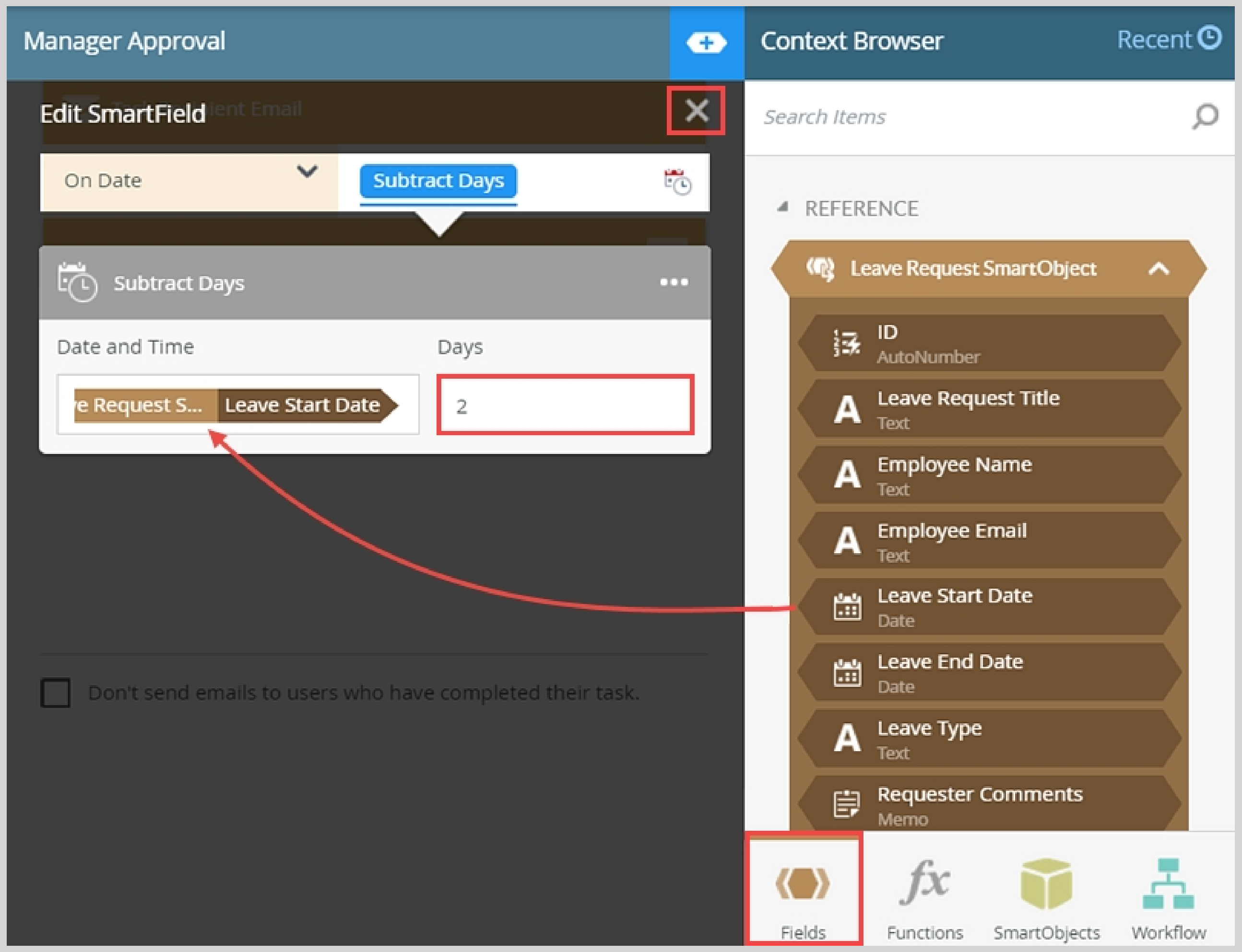
AND TIME node. Drag a **Subtract Days** function into the **Drop** box.



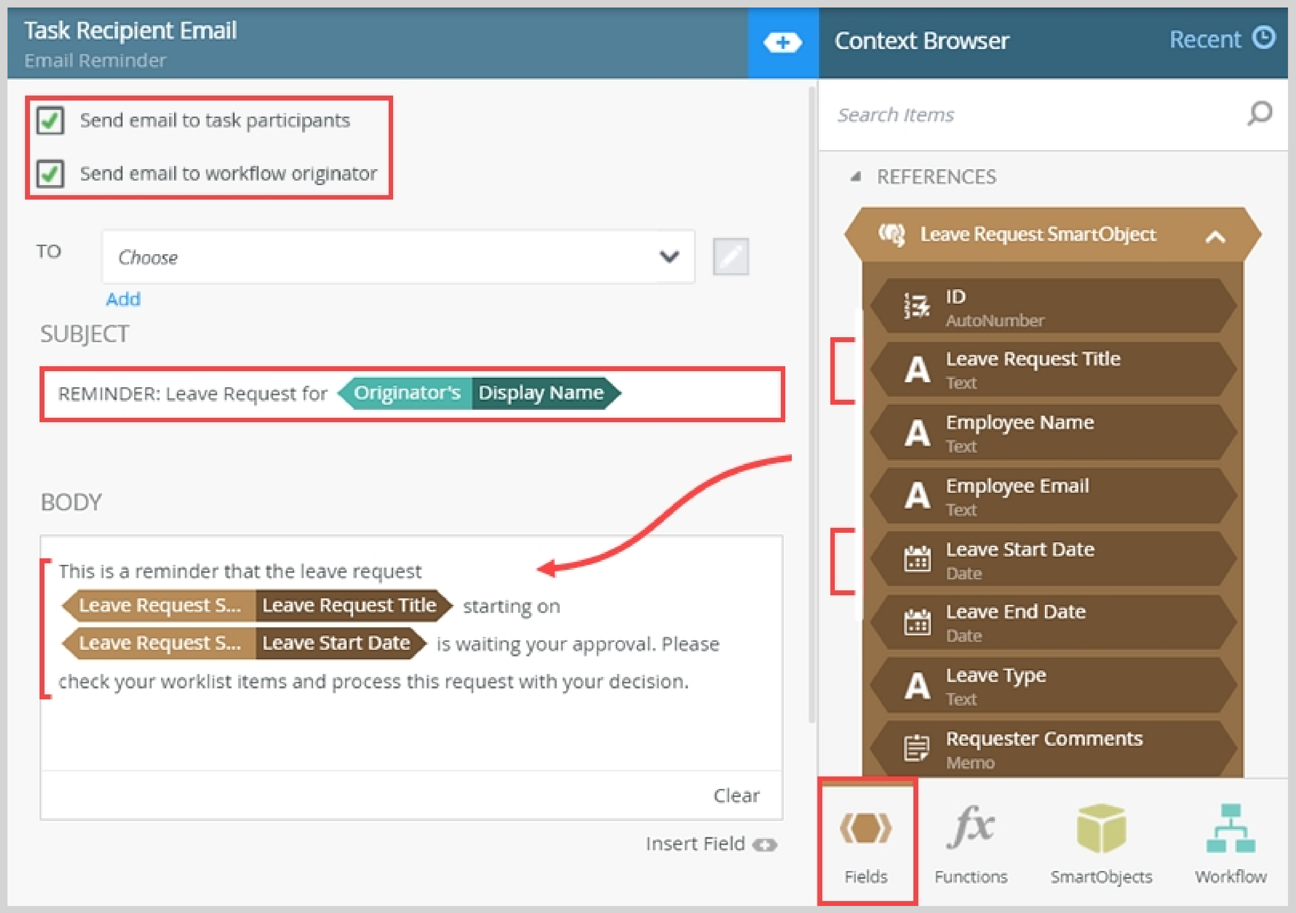
c. The Edit SmartField pane opens. Click the **Fields** tab and expand the **Leave Request SmartObject** reference. Drag the **Leave Start Date** into the **Date and Time** field. For the **Days** value, enter

2

then **exit** the SmartField editor. (Click the "X" in the upper right corner of the editor to exit it.) In this step, you configured a function that takes the Leave Start Date and subtracts two days from that date. (Remember this is the Subtract Days function, which is why you enter a number for the days to subtract.) This creates a "new" date for the reminder to fire.



2. Configure the reminder **recipients** as the **workflow originator** and **task participants**. The reminder email routes to the requester and the requester's manager. The requester can follow up with the manager to keep the workflow process flowing. For the subject line and message body, use variables from the **Context Browser** to personalize the reminder message.



Task Recipient Email
Email Reminder

☑ Send email to task participants

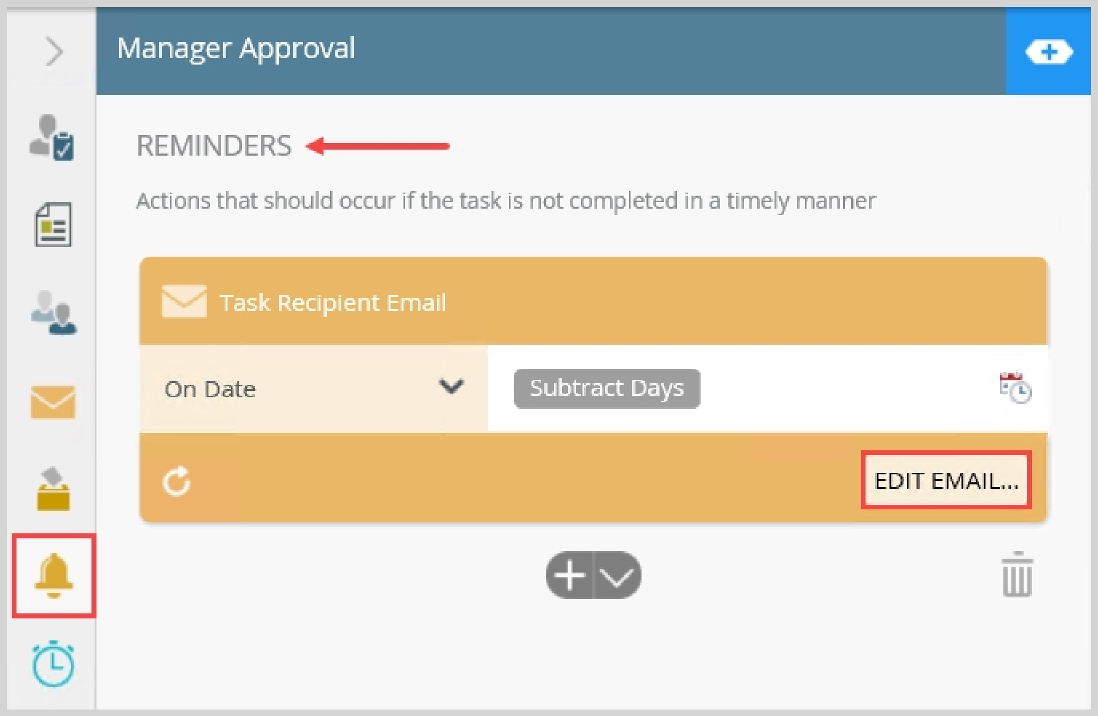
☑ Send email to workflow originator

TO: Choose

SUBJECT: REMINDER: Leave Request for Originator's Display Name

BODY: This is a reminder that the leave request Leave Request S... Leave Request Title starting on Leave Request S... Leave Start Date is waiting your approval. Please check your worklist items and process this request with your decision.

a. You will send the reminder email to the requester (originator) and the approving manager. This way, the requester can follow up with the manager to keep the workflow moving.
Still on the REMINDERS screen, click the **EDIT EMAIL** link.

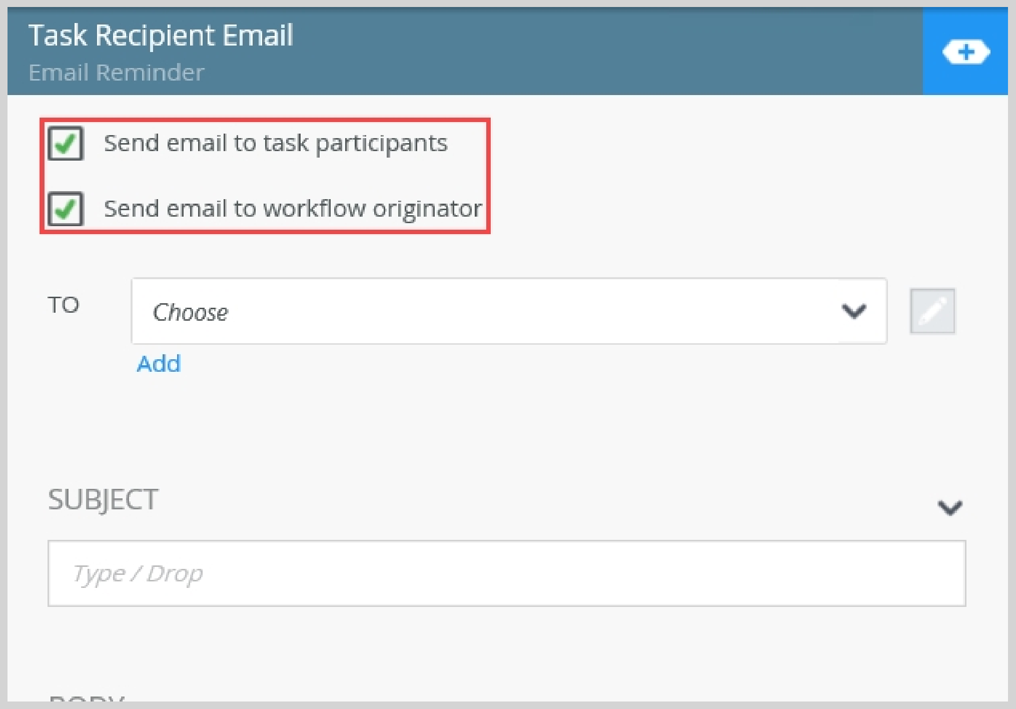


Manager Approval

REMINDERS

Actions that should occur if the task is not completed in a timely manner

b. CHECK the box to **Send email to workflow originator**. The box to **Send email to task participants** is already checked by default. The reminder email now routes to the leave requester (originator), in addition to the originator's manager.


Task Recipient Email
Email Reminder

☑ Send email to task participants

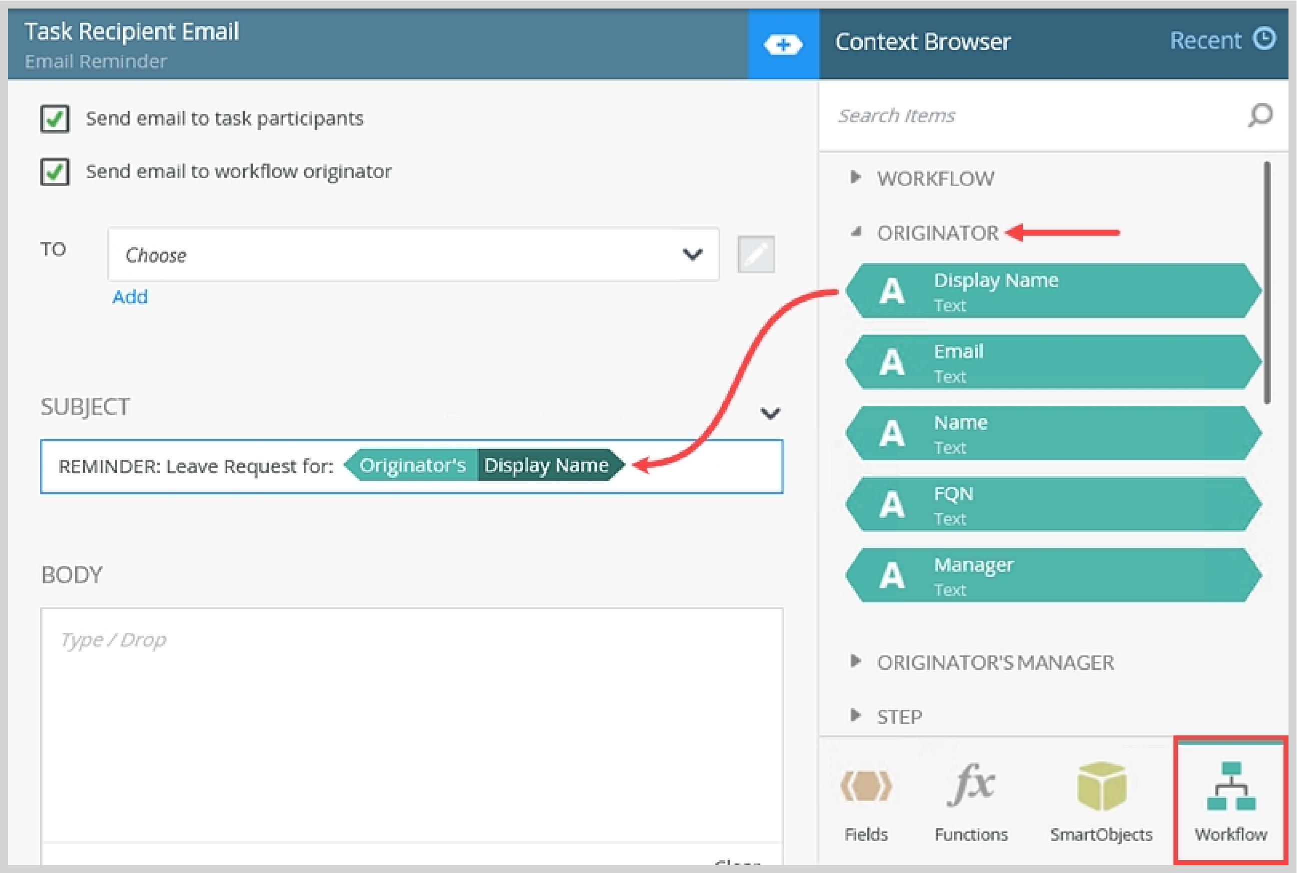
☑ Send email to workflow originator

TO Choose

SUBJECT

c. For the subject line, enter
REMINDER: Leave Request for [Originator Name]
then click the **Workflow** tab and expand the ORIGINATOR node. Drag the **Display Name** to the end of the subject line. Once again, this is a system value. K2 stores information about the current user, such as name, email, manager, etc., as system values that you can use through the workflow con-

figuration.



Task Recipient Email
Email Reminder

☑ Send email to task participants

☑ Send email to workflow originator

TO: Choose
Add

SUBJECT

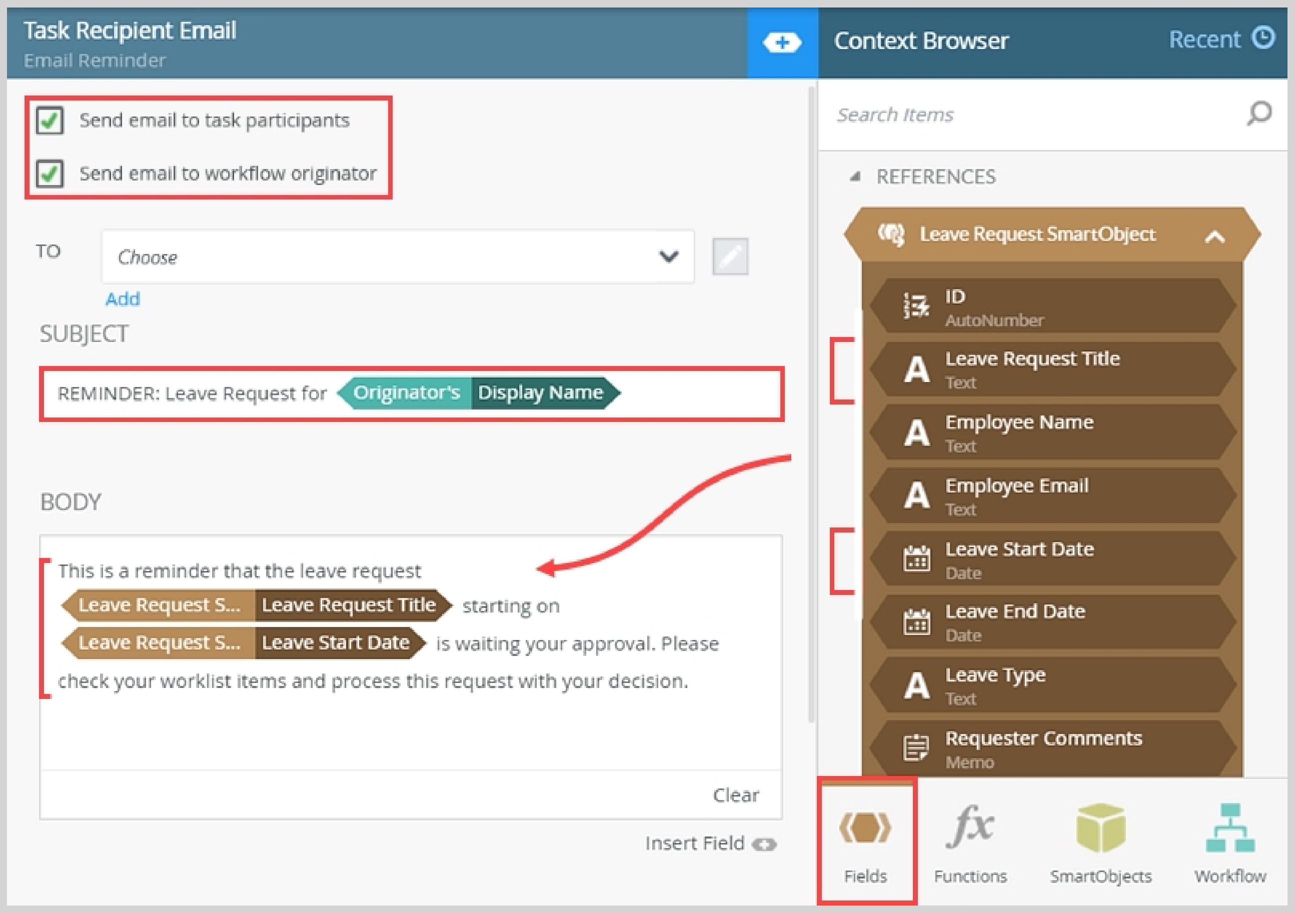
REMINDER: Leave Request for: Originator's Display Name

BODY

Type / Drop

d. For the message body, use **Leave Request SmartObject** references to personalize the email. The references are found within the **Fields** tab. Message body:
This is a reminder that the Leave Request [Leave Request Title] *starting on* [Leave Start Date] *has not*

yet been approved. Please check your worklist items and process this request with your decision.



Task Recipient Email
Email Reminder

☑ Send email to task participants

☑ Send email to workflow originator

TO: Choose

SUBJECT: REMINDER: Leave Request for Originator's Display Name

BODY: This is a reminder that the leave request Leave Request S... Leave Request Title starting on Leave Request S... Leave Start Date is waiting your approval. Please check your worklist items and process this request with your decision.

e. Collapse the **Escalation Email** pane, then collapse the **Configuration Panel**.

Review

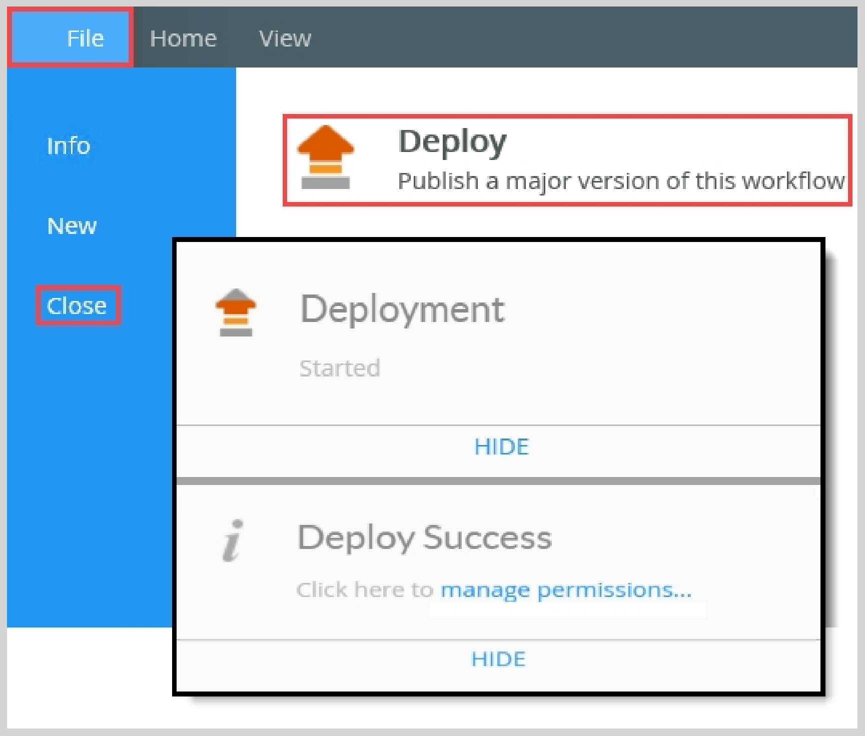
In this step, you added an reminder that sends an email reminder to the requester and the approving manager. The reminder fires two days before the Leave Start Date if the manager has not responded to the request. You personalized the email to include details about the request for the recipient's convenience. reminders are an effective tool for keeping your workflows running smoothly and on time. Be careful about sending too many reminders as you do not want to crowd the recipient's in-box!

Next Step: 8. Deploy the Leave Request Workflow

8. Deploy the Leave Request Workflow

Because you changed the Leave Request Workflow, you must redeploy it to the K2 server. Deploying the workflow publishes it and makes the current version available to your users. You must redeploy a workflow each time you make any changes.

1. **Deploy** the Leave Request Workflow.
   a. **Deploy** the workflow. (**File** > **Deploy**)
   b. After you see the **Deploy Success** dialog, return to the K2 Designer. (**File** > **Close**) Since you set permissions in the basic version tutorial, there are no other actions for this page.



Deployment

Started

Deploy Success

Click here to manage permissions...

Review

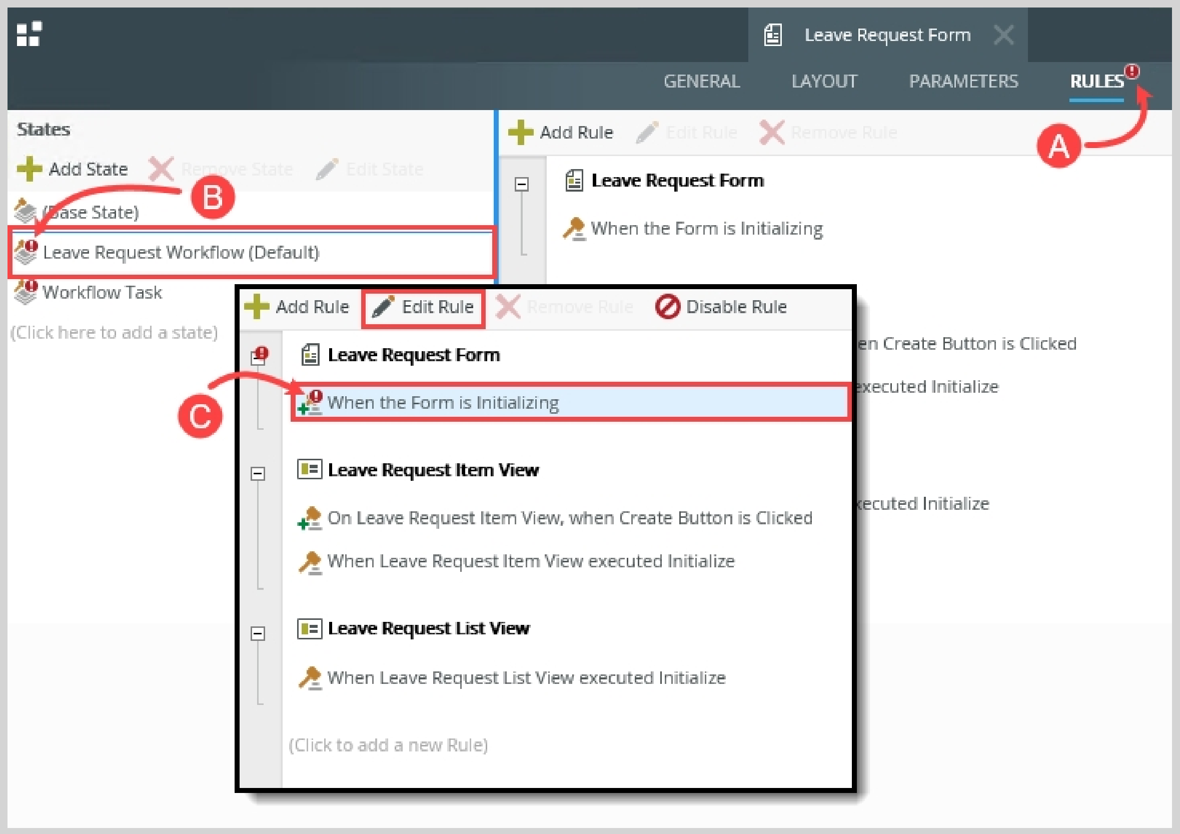
In this step, you deployed the Leave Request Workflow to the K2 server. The changes you made to the workflow are now ready for use! In the next step, you will edit the rules on the Leave Request Form to enable the Approver Comments control for the approving manager. (Remember, when you added the control to the item view, you made it read-only by default. Now you will enable it for the manager.)

Next Step: 9. Edit a Rule to Enable the Approver Comments for the Workflow Task State

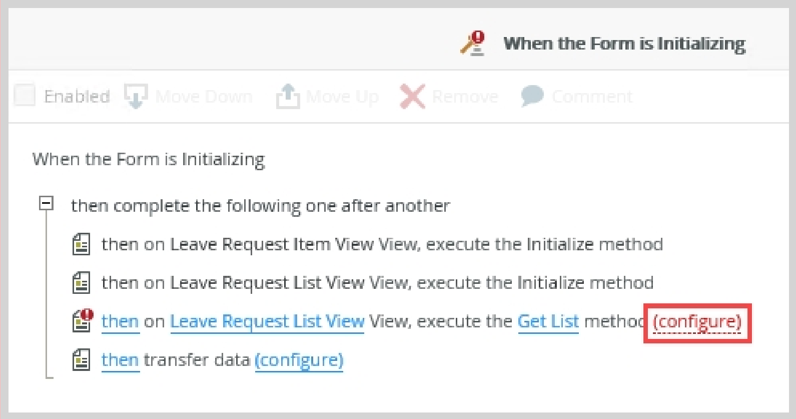
9. Edit a Rule to Enable the Approver Comments for the Workflow Task State

Currently, the *Approver Comments* text area box is a read-only control by default. In this step, you will edit the approving manager's state and enable the control. The manager can then enter any questions or comments they may have.

1. Check out, then edit the **Leave Request Form**. Edit the **form initializing** rule for the **Workflow Task** state. You are editing the rule that fires when the form first loads for the Manager Approval step.

> **Caution**
> **<u>You may see a red warning badge next to the RULES tab</u>**. This warning appears because you deleted the **Request Status** control from the item view and it is still bound to a rule or rules. Before continuing, switch to the **RULES** screen, then locate the **state** that also has a red warning badge. Click the state to expose its rules. **Edit** the offending rule.
>
> 
>

Leave Request Form
> GENERAL
> LAYOUT
> PARAMETERS
> RULES
> A
> B
> C
> States
> Add State
> Remove State
> Edit State
> (Base State)
> Leave Request Workflow (Default)
> Workflow Task
> (Click here to add a state)
> Add Rule
> Edit Rule
> Remove Rule
> Disable Rule
> Leave Request Form
> When the Form is Initializing
> Leave Request Item View
> On Leave Request Item View, when Create Button is Clicked
> When Leave Request Item View executed Initialize
> Leave Request List View
> When Leave Request List View executed Initialize
> (Click to add a new Rule)
> en Create Button is Clicked
> executed Initialize
> xecuted Initialize
>
> Click the red **(configure)** links (there may be more than one).



When the Form is Initializing

Enabled Move Down Move Up Remove Comment

When the Form is Initializing

then complete the following one after another

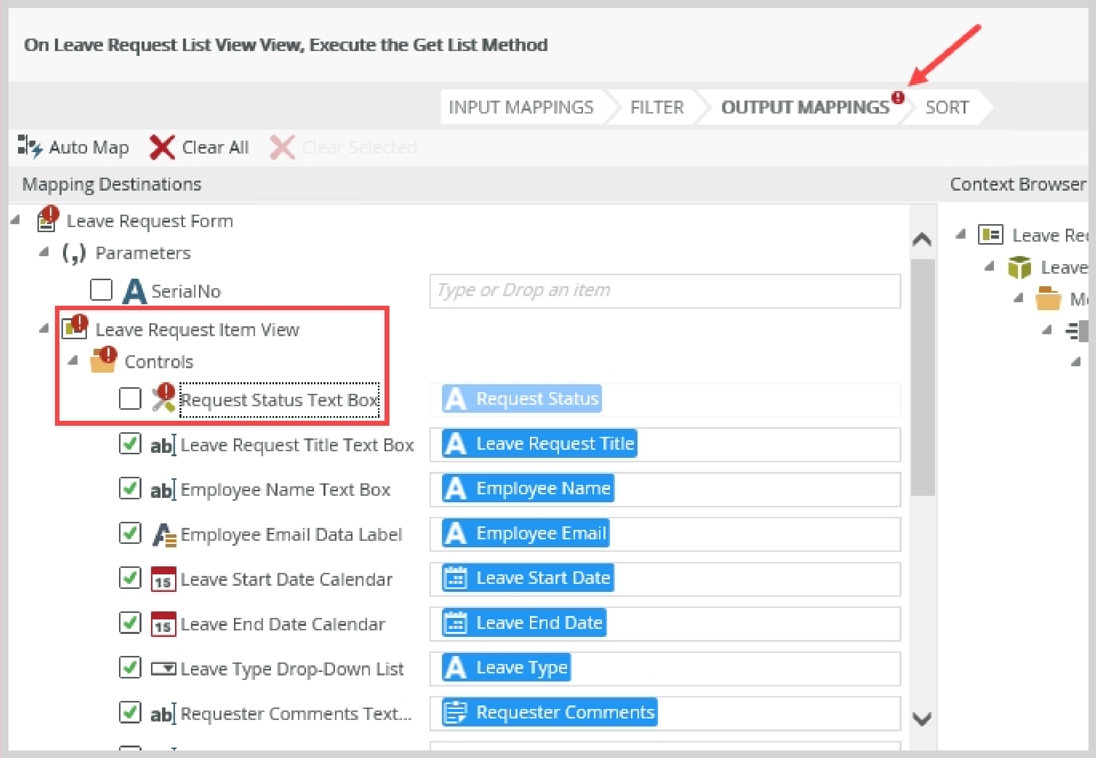
then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize method

then on Leave Request List View View, execute the Get List method (configure)

then transfer data (configure)

Most likely, the offending setting is in the OUTPUT MAPPINGS. Click the **OUTPUT MAPPINGS** tab. **UNCHECK** the box for the Request Status property.



On Leave Request List View View, Execute the Get List Method

INPUT MAPPINGS | FILTER | OUTPUT MAPPINGS | SORT

Auto Map | Clear All | Clear Selected

Mapping Destinations	
Leave Request Form	
Parameters	
□ SerialNo	Type or Drop an item
Leave Request Item View	
Controls	
□ Request Status Text Box	Request Status
☑ Leave Request Title Text Box	Leave Request Title
☑ Employee Name Text Box	Employee Name
☑ Employee Email Data Label	Employee Email
☑ Leave Start Date Calendar	Leave Start Date
☑ Leave End Date Calendar	Leave End Date
☑ Leave Type Drop-Down List	Leave Type
☑ Requester Comments Text...	Requester Comments

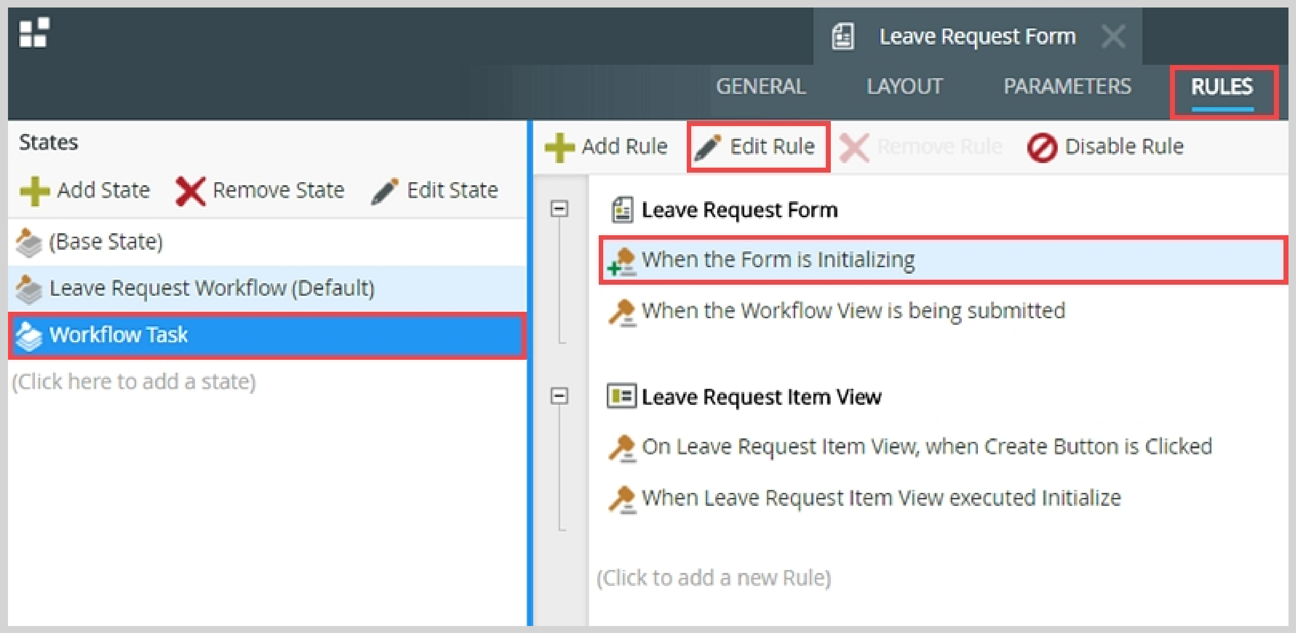
Click **FINISH** to save the setting. Click **OK** to save the rule.

Repeat this step for all states that display a red warning badge.

a. Recall that states represent the form's behavior and configuration for a particular step, usually a Task step that assigns a task to a recipient. You can configure rules to customize the form's behavior for the recipient. In this case, when added the Approver Comments control to the Leave Request Item View, you made the control read-only so that the leave requester could not change it. Now, you will enable the control for the approving manager, so that they can enter questions and comments if they need to. In K2 Designer, right-click the **Leave Request Form** and check it out (if it is not already), then **Edit** it.

Caution
Before continuing, check to see if there is a **red warning badge** next to the **RULES** tab. If so, go back and follow the instructions in the red caution box above before proceeding.

b. Click the **RULES** tab in the breadcrumb bar. Highlight the **Workflow Task** state, then highlight the **When the Form is Initializing** rule. Click **Edit Rule**. The Workflow Task state represents the form configuration and behavior for the Manager Approval step. The form initializing rule fires when the form first loads.



(Base State)
Leave Request Workflow (Default)
Workflow Task
(Click here to add a state)

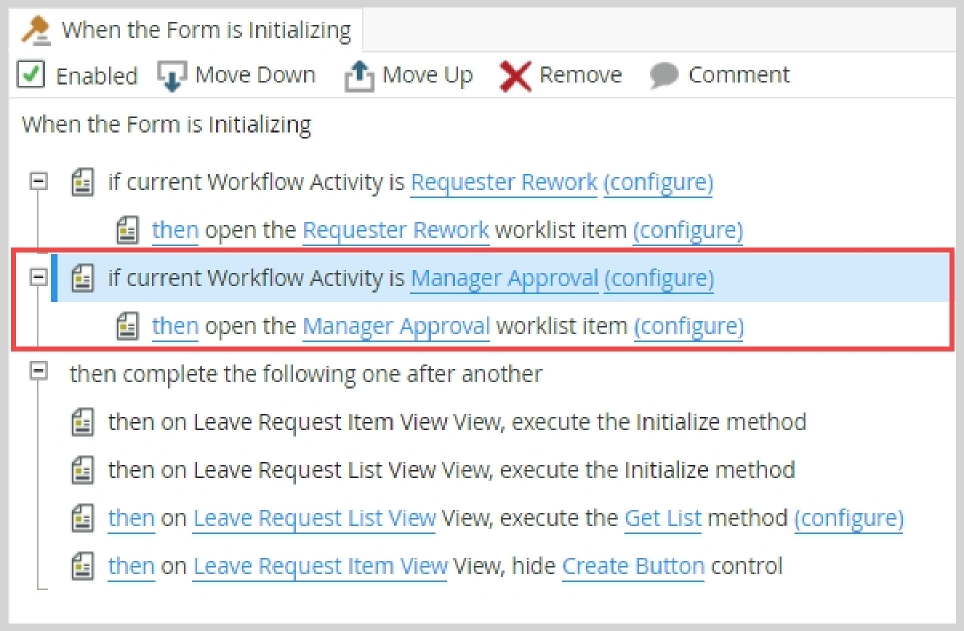
Leave Request Form
When the Form is Initializing
When the Workflow View is being submitted

Leave Request Item View
On Leave Request Item View, when Create Button is Clicked
When Leave Request Item View executed Initialize

(Click to add a new Rule)

2. In the **if current Workflow Activity is Manager Approval** condition, add an action to **set a View control's properties** on the **Leave Request Item View** that enables the **Approver Comments Text Area** control. Here, you are enabling the Approver Comments control for the approving manager.

   a. In the rule definition pane, locate the condition that says **if current Workflow Activity is Manager Approval**. Notice that this condition already has an action to open the Manager Approval worklist item. K2 added this condition and action when you configured the Manager Approval step. You want to add an action at this point to enable the Approver Comments text area box.

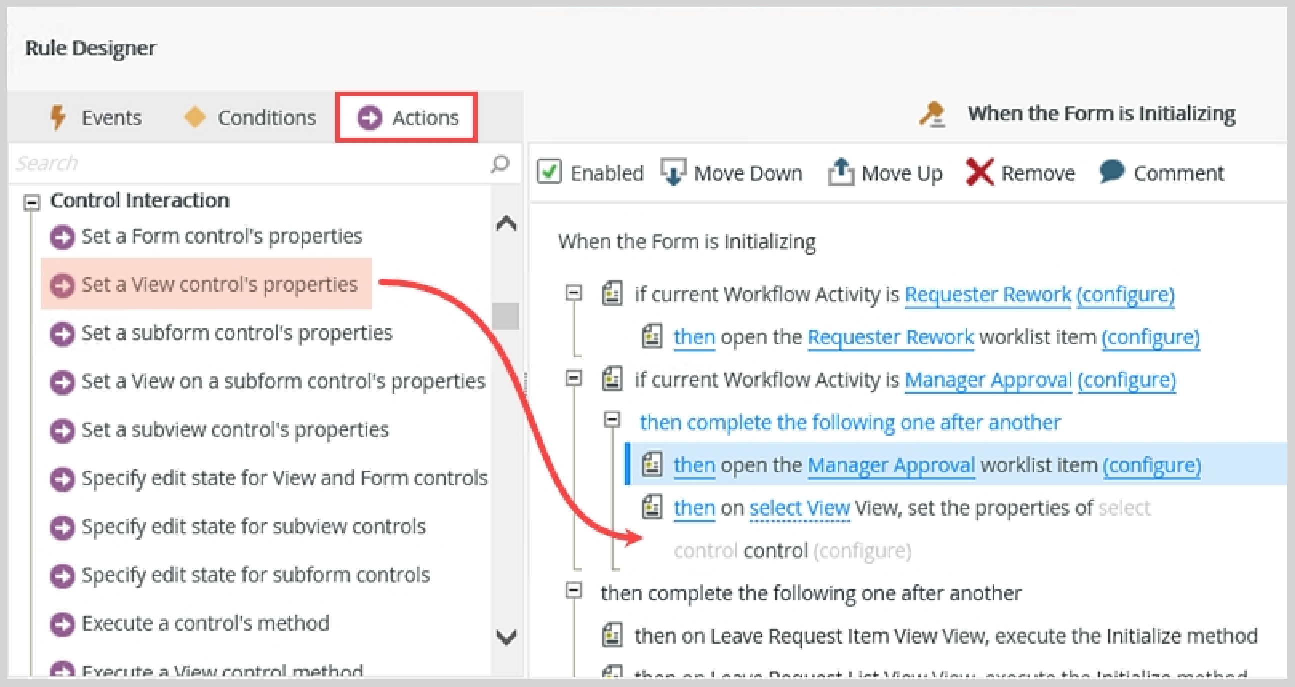


When the Form is Initializing

if current Workflow Activity is Requester Rework (configure)
then open the Requester Rework worklist item (configure)
if current Workflow Activity is Manager Approval (configure)
then open the Manager Approval worklist item (configure)
then complete the following one after another
then on Leave Request Item View View, execute the Initialize method
then on Leave Request List View View, execute the Initialize method
then on Leave Request List View View, execute the Get List method (configure)
then on Leave Request Item View View, hide Create Button control

b. Confirm the **Actions** tab is active. Click the **then open the Manager Approval worklist item** action to highlight it. In the **Actions** pane, scroll down until you see the **Control Interaction** heading. Click the **Set a View control's properties** action to add it to the Workflow Activity condition, just below the open worklist item action.

> **Note**
> The reason you select the worklist item action first is to tell K2 where to insert the new action. The order of events, conditions, and actions is very important as K2 follows them in a linear fashion. If you did not select the worklist item action, K2 would have placed the new action at the bottom of the rule definition pane. You can always move actions into the correct order by selecting them, then using the control arrows to move them up or down. (After you select an action, simply hover over it to expose the control arrows.)


Rule Designer
Events
Conditions
Actions
Search
Control Interaction
Set a Form control's properties
Set a View control's properties
Set a subform control's properties
Set a View on a subform control's properties
Set a subview control's properties
Specify edit state for View and Form controls
Specify edit state for subview controls
Specify edit state for subform controls
Execute a control's method
Execute a View control method
When the Form is Initializing
Enabled
Move Down
Move Up
Remove
Comment

When the Form is Initializing

if current Workflow Activity is Requester Rework (configure)

then open the Requester Rework worklist item (configure)

if current Workflow Activity is Manager Approval (configure)

then complete the following one after another

then open the Manager Approval worklist item (configure)

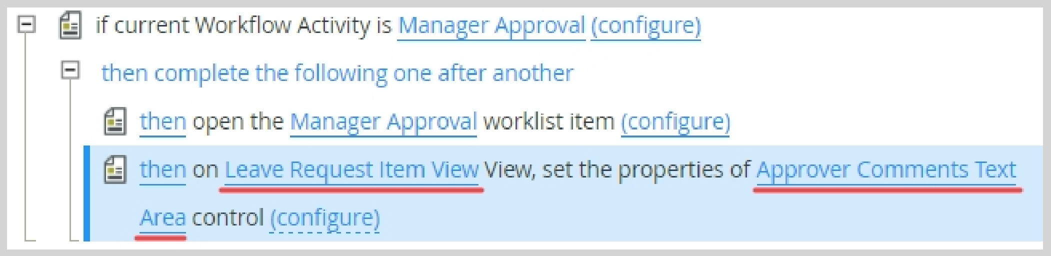
then on select View View, set the properties of select control control (configure)

then complete the following one after another

then on Leave Request Item View View, execute the Initialize method

then on Leave Request List View View, execute the Initialize method

c. Click the **select View** link and select **Leave Request Item View**. Click the **select control** link and select **Approver Comments Text Area**.



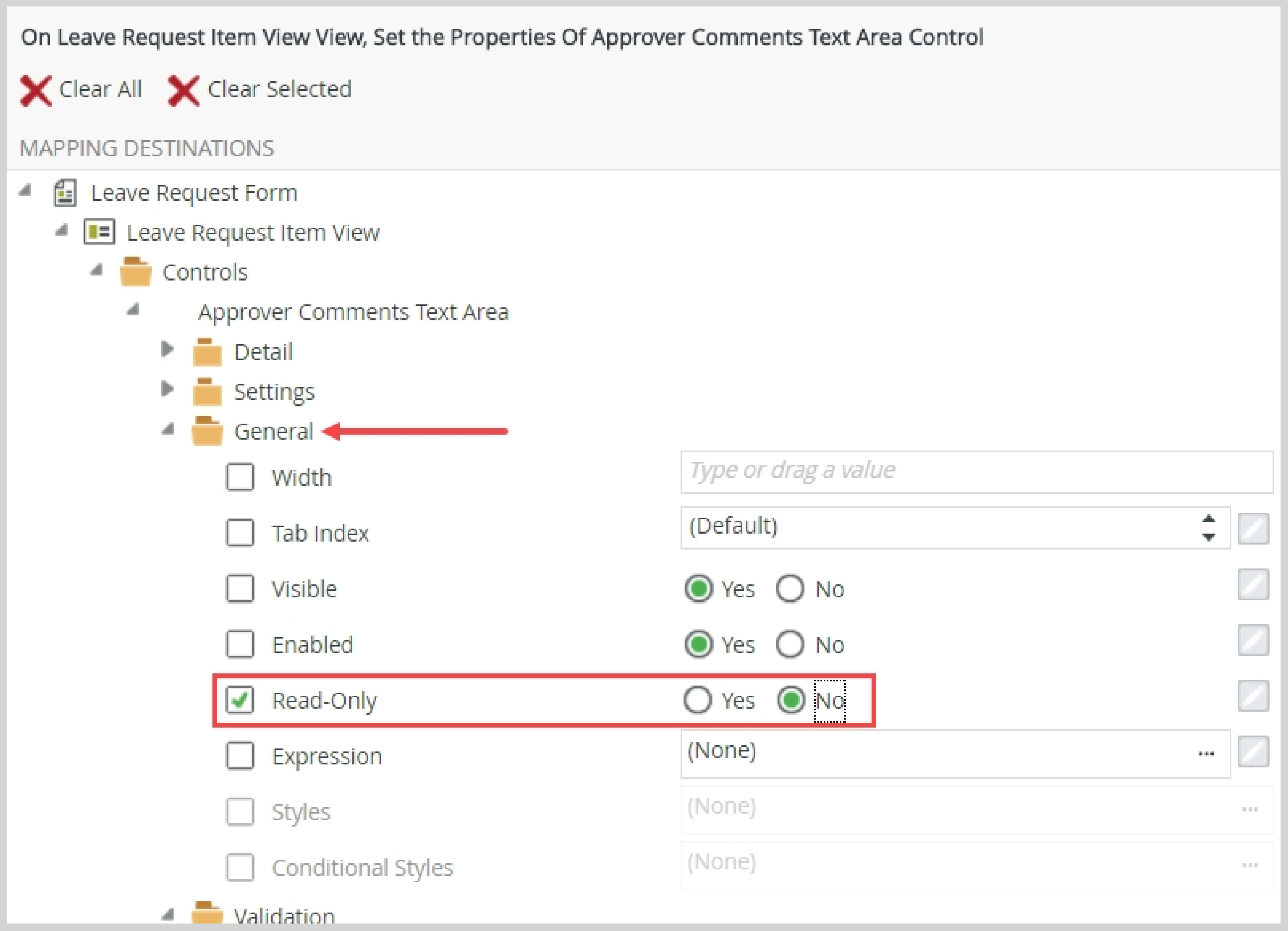
if current Workflow Activity is Manager Approval (configure)

then complete the following one after another

then open the Manager Approval worklist item (configure)

then on Leave Request Item View View, set the properties of Approver Comments Text Area control (configure)

d. Click the **(configure)** link. On the **Mapping Destinations** screen, scroll down until you can see the **General** heading. Change the **Read-Only** setting from Yes to **No**. By turning off the read-only setting, the manager is able to add comments when they open the form. Click **OK**, then **OK** once again to return to the K2 Designer. (Do not close out of the form just yet.)



On Leave Request Item View View, Set the Properties Of Approver Comments Text Area Control

□ Width	Type or drag a value	
□ Tab Index	(Default)	
□ Visible	● Yes ○ No	
□ Enabled	● Yes ○ No	
☑ Read-Only	○ Yes ● No	
□ Expression	(None)	...
□ Styles	(None)	...
□ Conditional Styles	(None)	...

Review

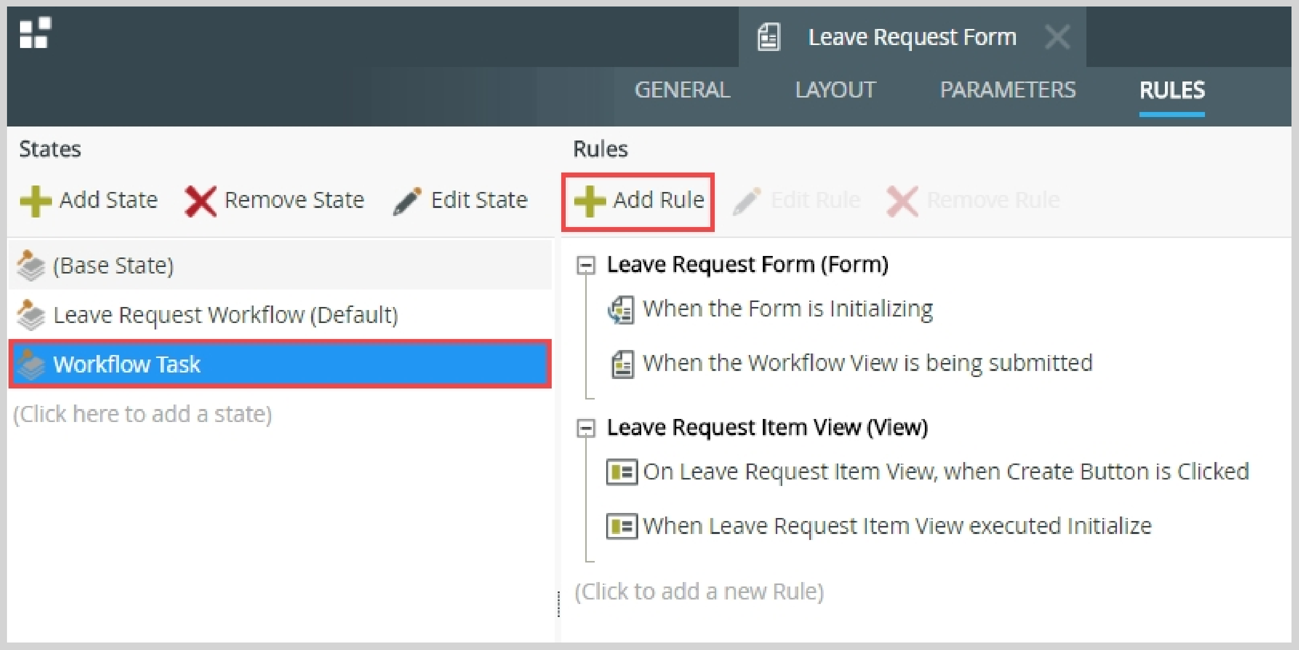
In this step, you edited the Workflow Task state. This state represents the form behavior for the Manager Approval step. You enabled the Approver Comments text area box so that the manager can enter any questions or comments they may have. In the next step, you will add an action that saves the form fields to the SmartObject after the manager submits their decision.

Next Step: 10. Add a Save Method to Update the SmartObject with Manager Changes

10. Add a Save Method to Update the SmartObject with Manager Changes

In this step you will add a new rule to the Workflow Task state. (This is the form state for the Manager Approval step.) The new rule fires when the approving manager submits their decision (such as Approved, Denied or Rework). The rule action saves the form fields into the Leave Request SmartObject with any changes the manager made.

1. Add a new rule to the Workflow Task state. Add the event **After the Workflow action was submitted**. The rule fires just after the manager submits their decision.
   a. On the RULES screen, select the **Workflow Task** state (if it is not already), then click **Add Rule**.



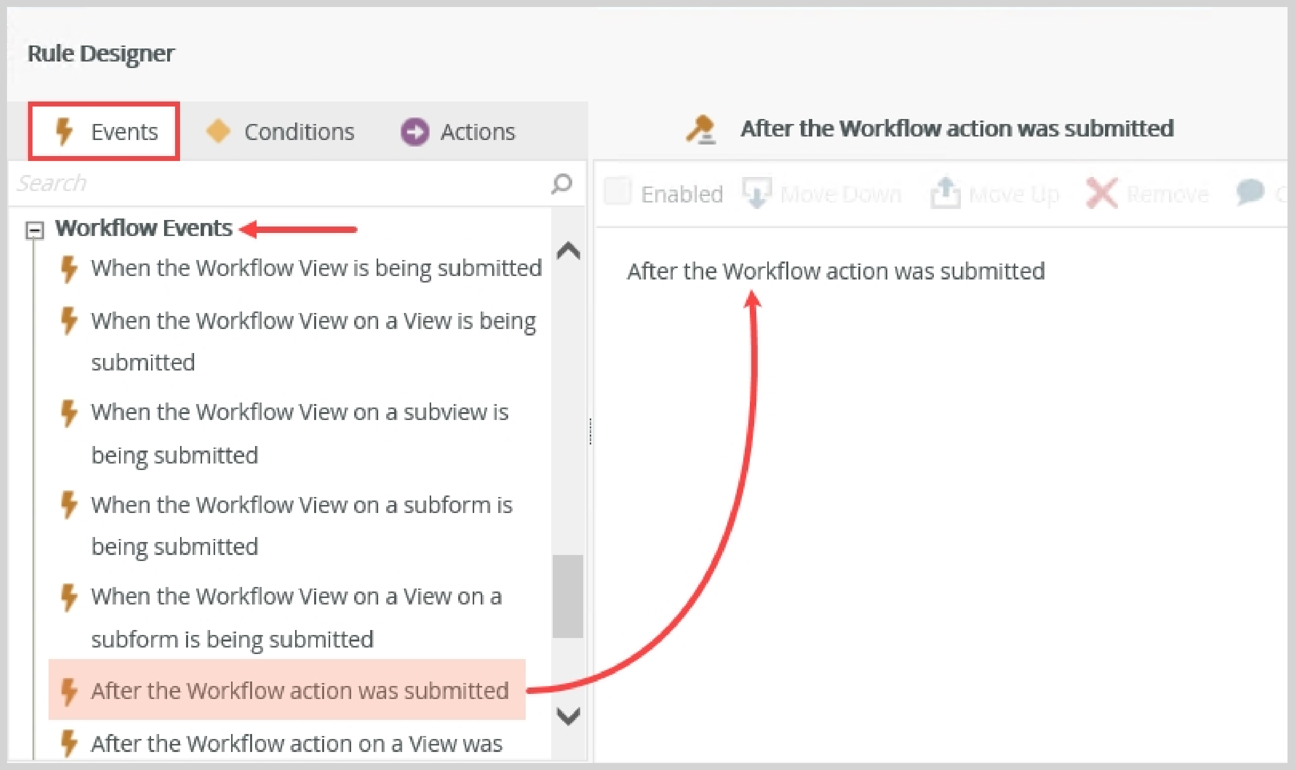
(Base State)
Leave Request Workflow (Default)
Workflow Task
(Click here to add a state)

Leave Request Form (Form)
- When the Form is Initializing
- When the Workflow View is being submitted

Leave Request Item View (View)
- On Leave Request Item View, when Create Button is Clicked
- When Leave Request Item View executed Initialize

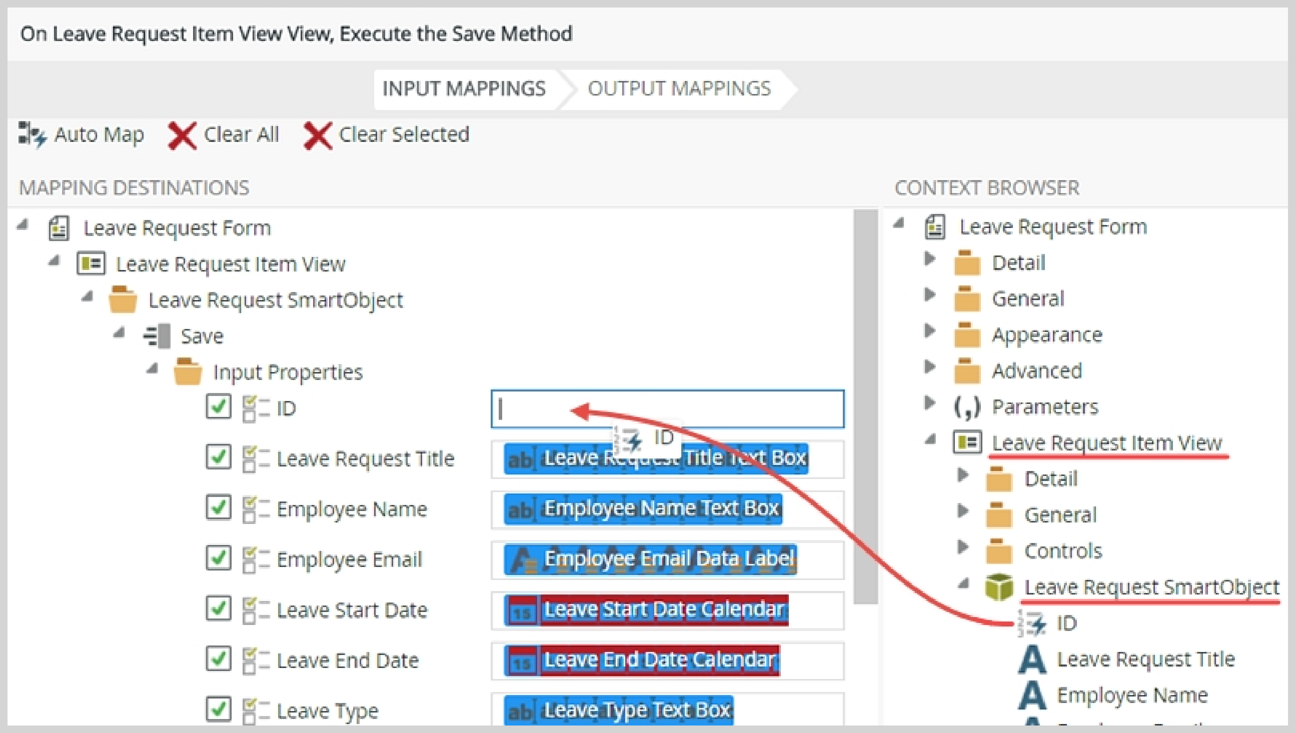
(Click to add a new Rule)

b. Confirm the **Events** tab is active. Scroll down until you see the **Workflow Events** heading. Locate and click the **After the Workflow action was submitted** event to add it to the rule definition pane.



Rule Designer
Events
Conditions
Actions
Search
Workflow Events
When the Workflow View is being submitted
When the Workflow View on a View is being submitted
When the Workflow View on a subview is being submitted
When the Workflow View on a subform is being submitted
When the Workflow View on a View on a subform is being submitted
After the Workflow action was submitted
After the Workflow action on a View was
After the Workflow action was submitted
Enabled
Move Down
Move Up
Remove
After the Workflow action was submitted

2. Add an **Execute a View method** action to call the **Save** method of the **Leave Request Item View**. You may **auto-map** the input properties, but you MUST delete the auto-mapped ID value and replace it with the **Leave Request Item View > Leave Request SmartObject** > **ID** property so that K2 knows you are referencing the current record. The ID generated by auto-mapping the input properties does not match the ID of the record you want to update. Because of this, K2 may throw an error, as it does not know which record to update. You must replace the auto-mapped ID with the ID of the current record.



On Leave Request Item View View, Execute the Save Method

MAPPING DESTINATIONS

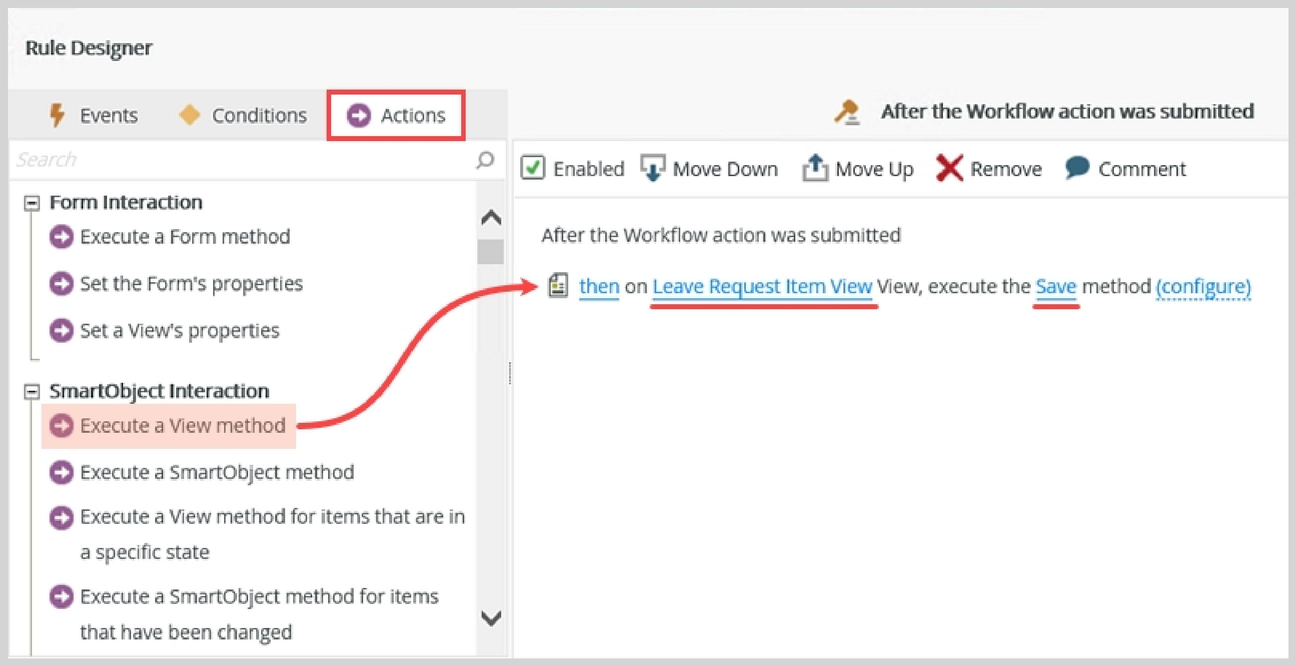
- Leave Request Form
  - Leave Request Item View
    - Leave Request SmartObject
      - Save
        - Input Properties

Input Property	Mapping
☑ ID	ID
☑ Leave Request Title	Leave Request Title Text Box
☑ Employee Name	Employee Name Text Box
☑ Employee Email	Employee Email Data Label
☑ Leave Start Date	Leave Start Date Calendar
☑ Leave End Date	Leave End Date Calendar
☑ Leave Type	Leave Type Text Box

CONTEXT BROWSER

- Leave Request Form
  - Detail
  - General
  - Appearance
  - Advanced
  - (,) Parameters
  - Leave Request Item View
    - Detail
    - General
    - Controls
    - Leave Request SmartObject
      - ID
      - Leave Request Title
      - Employee Name

a. Now that you have the event set, you will add the action that updates the current leave request record with any changes the manager made.
Click the **Actions** tab so that it is active. Click to add the **Execute a View method** action to the rule definition pane. Click the **select View** link and select **Leave Request Item View**. Click the **select method** link and select **Save**. The **save** action updates an existing record in the Leave Request SmartObject. In this case, you want to save any changes the approving manager made to the leave request. Next, you will use the auto-map feature, which saves time by mapping the input properties for you.



Rule Designer

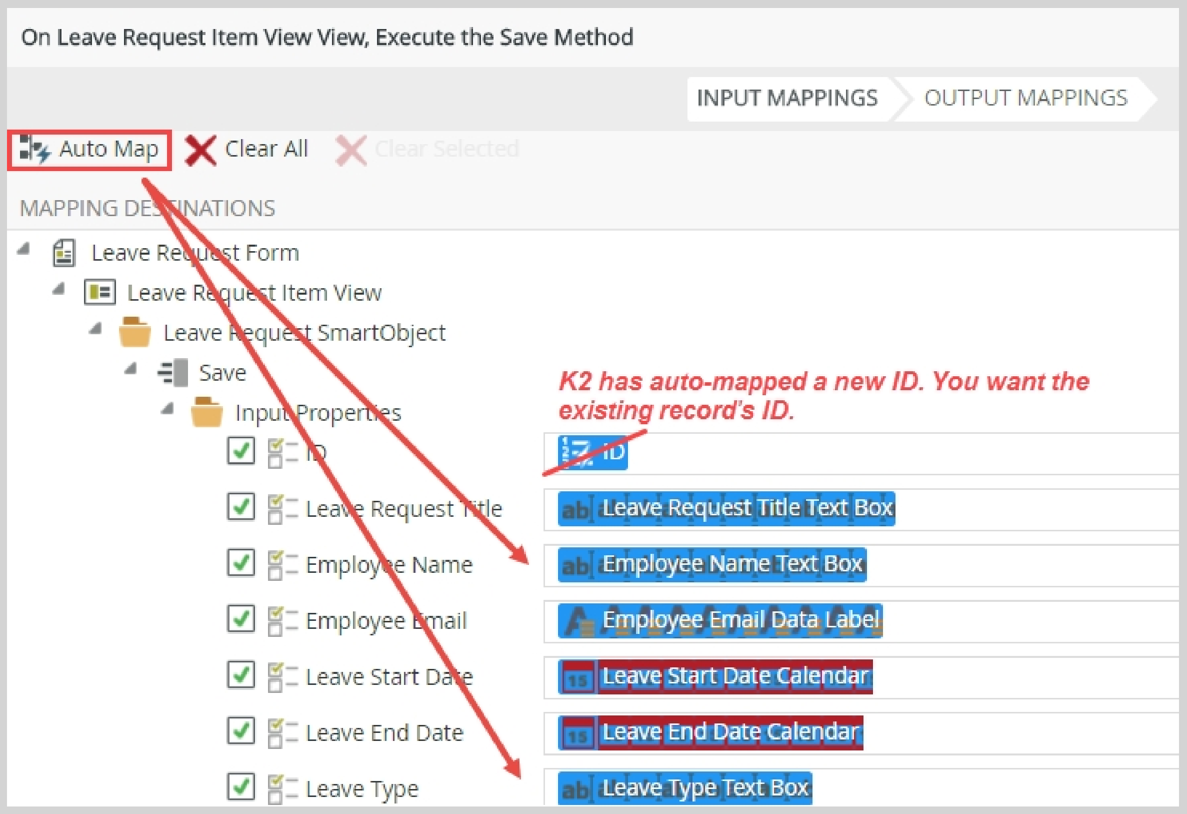
After the Workflow action was submitted

- Form Interaction
  - Execute a Form method
  - Set the Form's properties
  - Set a View's properties
- SmartObject Interaction
  - Execute a View method
  - Execute a SmartObject method
  - Execute a View method for items that are in a specific state
  - Execute a SmartObject method for items that have been changed

After the Workflow action was submitted

then on Leave Request Item View View, execute the Save method (configure)

b. Click the **(configure)** link. Click **Auto-Map**. K2 auto-maps the input properties (think form fields) to their corresponding SmartObject properties (Leave Request SmartObject properties). There is one change you must make before continuing and that is to replace the auto-mapped ID with the ID of the current record. Because you are updating a record, the record already exists. You must use the ID of the existing record. The auto-mapped ID creates new record, which is not what you want. So, you will replace the auto-mapped ID with the existing record ID. Do not skip this step or you likely experience errors later on!



On Leave Request Item View View, Execute the Save Method

Leave Request Form
Leave Request Item View
Leave Request SmartObject
Save
Input Properties

☑ ID	ID
☑ Leave Request Title	Leave Request Title Text Box
☑ Employee Name	Employee Name Text Box
☑ Employee Email	Employee Email Data Label
☑ Leave Start Date	Leave Start Date Calendar
☑ Leave End Date	Leave End Date Calendar
☑ Leave Type	Leave Type Text Box

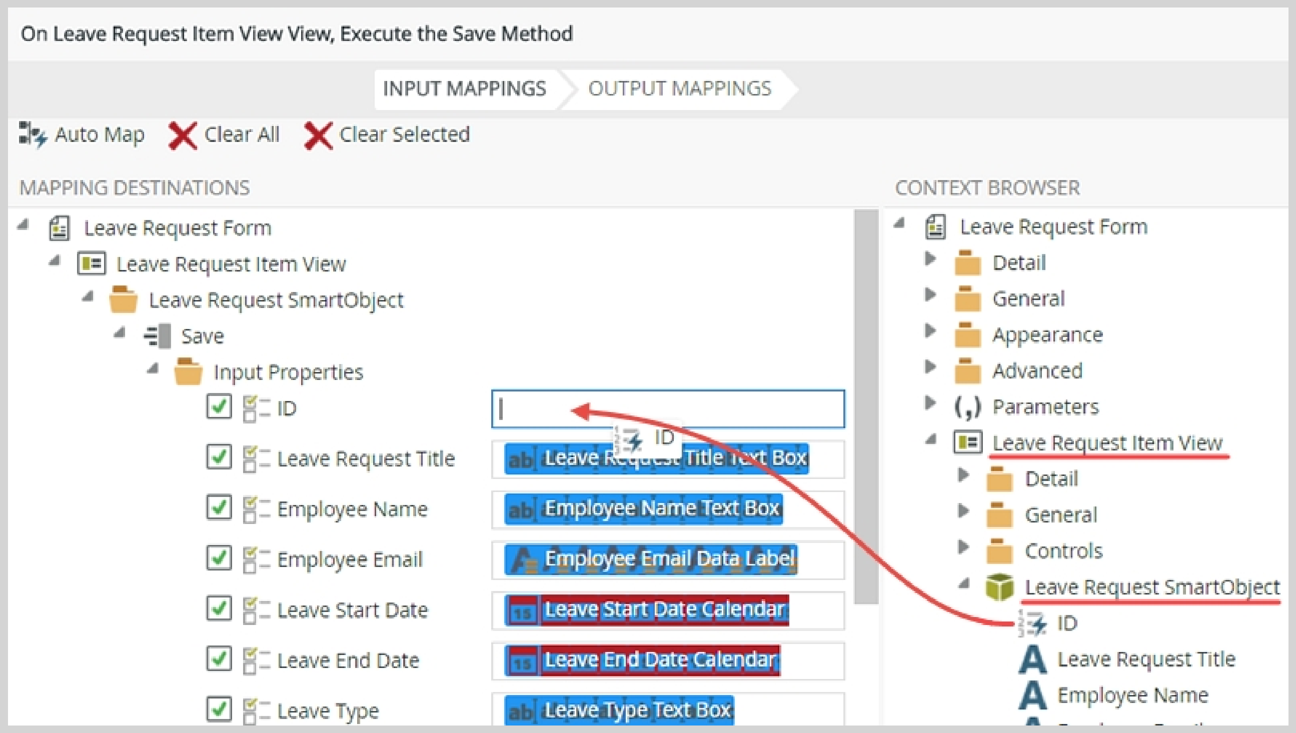
K2 has auto-mapped a new ID. You want the existing record's ID.

c. Click to the right of the auto-mapped **ID** value and backspace until you have removed it.

d. Now, you will add the SmartObject ID for the current record that you want to update. In the CONTEXT BROWSER, expand the **Leave Request Item View**. Expand the **Leave Request SmartObject.** Drag the SmartObject **ID** into the **Input Properties** > **ID** field.

> **Caution**
>
> **DO NOT SKIP THIS STEP!** Be sure to complete this step. You want to use the SmartObject ID so that K2 knows you are updating the current record. If you use the auto-mapped ID, K2 will not recognize the value as the current record and will create a new record instead, or possibly throw an error.

Click **FINISH**. Click **OK** to close the Rule Designer.


On Leave Request Item View View, Execute the Save Method

☑ ID	
☑ Leave Request Title	Leave Request Title Text Box
☑ Employee Name	Employee Name Text Box
☑ Employee Email	Employee Email Data Label
☑ Leave Start Date	Leave Start Date Calendar
☑ Leave End Date	Leave End Date Calendar
☑ Leave Type	Leave Type Text Box

Review

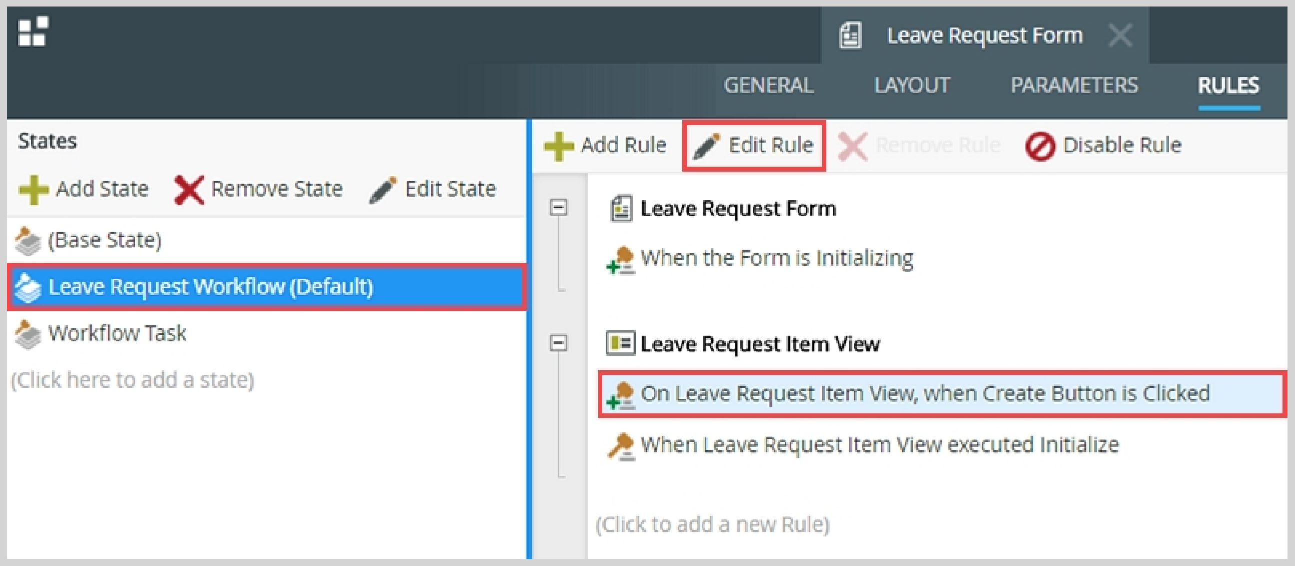
In this step, you added a new rule that saves changes the approving manager made back to the Leave Request SmartObject. You used the auto-mapping feature that saves time, especially if you have a large number of input properties to map. To make sure K2 updates the current *existing* record, you mapped the Leave Request SmartObject > ID to the ID input property. In the next step, you will add a clear method that clears the form fields after the leave request submits the form. This gives a visual indication that the form submitted successfully.

Next Step: 11. Add a Clear Method to Clear Form Fields After Submit

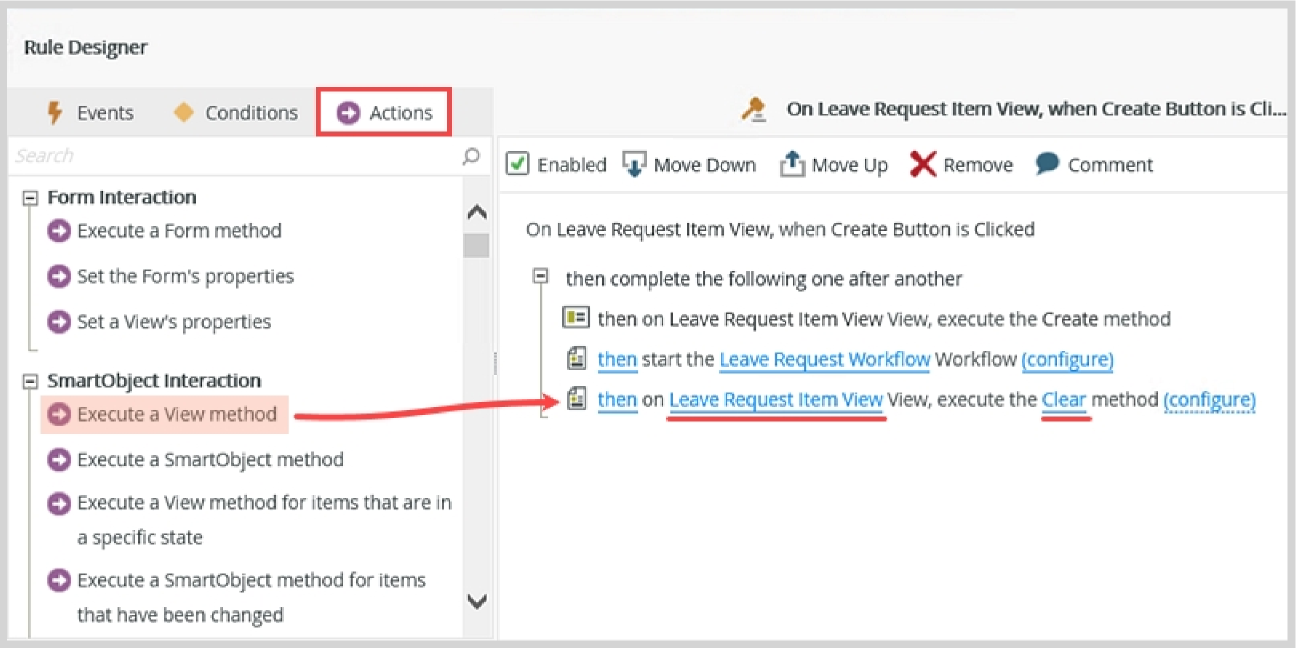
11. Add a Clear Method to Clear Form Fields After Submit

In this step, you will add an action that clears the form fields after you submit the form. By default, there is no visual indication that the form submitted. So, by clearing the form fields, you have a visual indication the form submitted. You also have a clean form if you want to submit another leave request. You will add an action to pre-populate the logged-in user's name and email when the form loads. This personalizes the user experience using system values. Last, you will add an action that updates the Leave Request List View with the new record just entered.

1. Edit the **Create Button is Clicked** rule for the **Leave Request Workflow (Default)** state. Add a view method action to **Clear** the **Leave Request Item View**. This step gives you a visual indication the form submitted successfully. Other options include adding a message box to indicate the form submitted or redirecting you to another location, a website for example.
    a. First, you will configure an action to clear the form entries after the form submits. This gives you a visual indication that the form submitted successfully. Still on the **RULES** screen, highlight the **Leave Request Workflow (Default)** state, then the **...Create Button is Clicked** rule. The Leave Request Workflow (Default) state is the form configuration and behavior the requester sees when they first submit a leave request. Click **Edit Rule**.


Leave Request Form
GENERAL
LAYOUT
PARAMETERS
RULES
States
Add State
Remove State
Edit State
(Base State)
Leave Request Workflow (Default)
Workflow Task
(Click here to add a state)
Add Rule
Edit Rule
Remove Rule
Disable Rule
Leave Request Form
When the Form is Initializing
Leave Request Item View
On Leave Request Item View, when Create Button is Clicked
When Leave Request Item View executed Initialize
(Click to add a new Rule)

    b. Confirm the **Actions** tab is highlighted, then click the **Execute a View method** action to add it to the rule definition pane. Click the **select View** link and select **Leave Request Item View**. Click the **select method** link and select **Clear**. You don't need to configure anything else. The action will clear out all the form fields, leaving you still on the form. Now, the form is ready for another leave request entry if needed.


Rule Designer
Events
Conditions
Actions
Search
Form Interaction
Execute a Form method
Set the Form's properties
Set a View's properties
SmartObject Interaction
Execute a View method
Execute a SmartObject method
Execute a View method for items that are in a specific state
Execute a SmartObject method for items that have been changed
On Leave Request Item View, when Create Button is Cli...
Enabled
Move Down
Move Up
Remove
Comment

On Leave Request Item View, when Create Button is Clicked

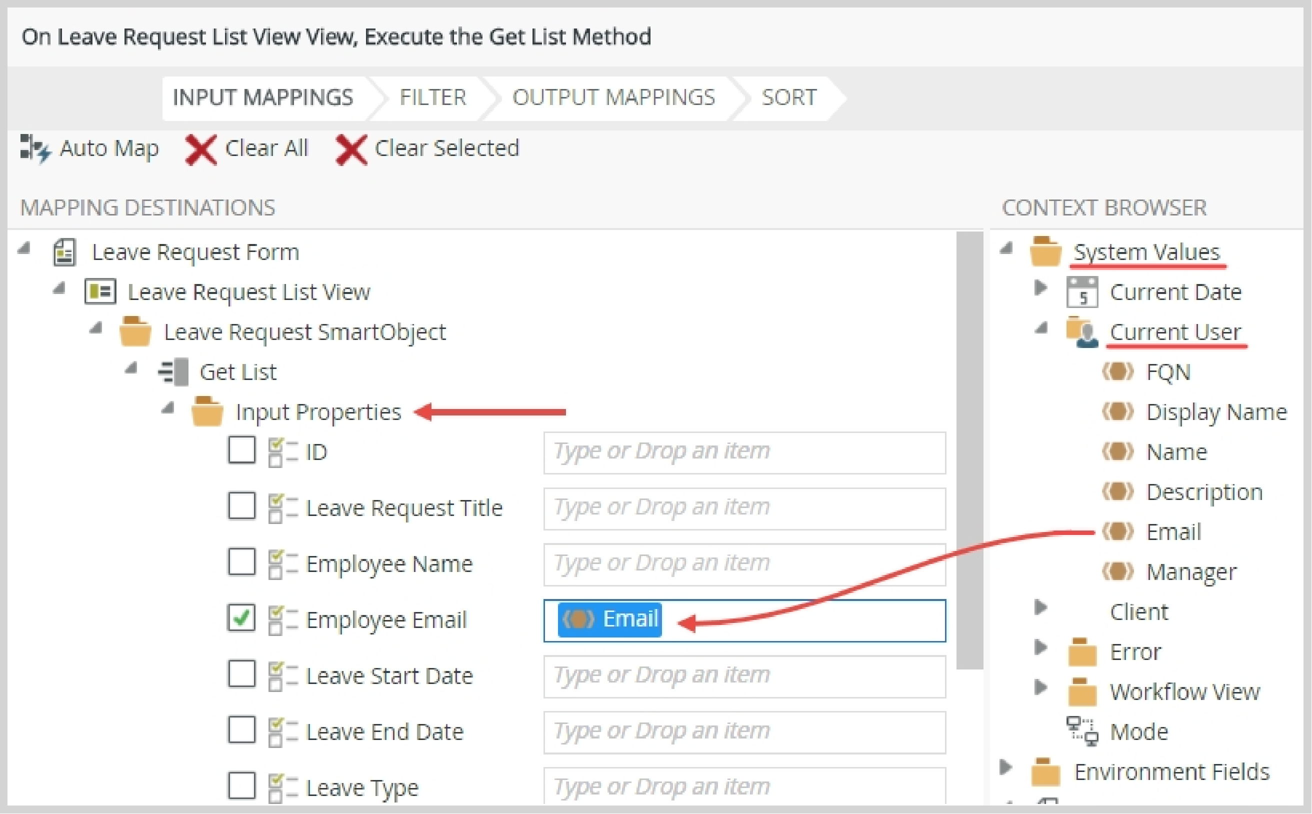
then complete the following one after another

then on Leave Request Item View View, execute the Create method

then start the Leave Request Workflow Workflow (configure)

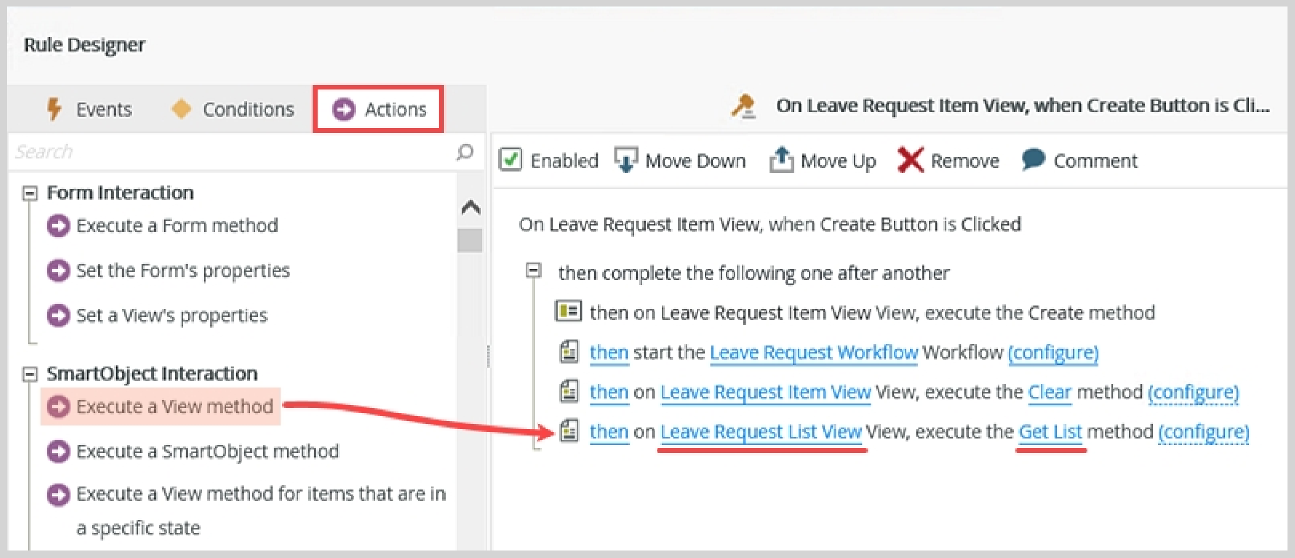
then on Leave Request Item View View, execute the Clear method (configure)

2. Add another view method action to populate the **Leave Request List View** (**Get List**), using the **System Value** > **Current User** > **Email** for the input property. In this step, you are "refreshing" the list view so that it displays the leave request just submitted.



☐ ID	Type or Drop an item
☐ Leave Request Title	Type or Drop an item
☐ Employee Name	Type or Drop an item
☑ Employee Email	Email
☐ Leave Start Date	Type or Drop an item
☐ Leave End Date	Type or Drop an item
☐ Leave Type	Type or Drop an item

a. Next, you will update the Leave Request List View so that it displays the new record you just submitted. If you didn't add this action, the list view would not display the new record until you manually refreshed the page to reload the view. This action is forcing the view to reload behind-the-scenes. Click **Execute a View method** once again to add another instance to the rule definition pane. Click the **select View** link and select **Leave Request List View**. Click the **select method** link and select **Get List**. Click the **(configure)** link.



Rule Designer

On Leave Request Item View, when Create Button is Cli...

On Leave Request Item View, when Create Button is Clicked

then complete the following one after another

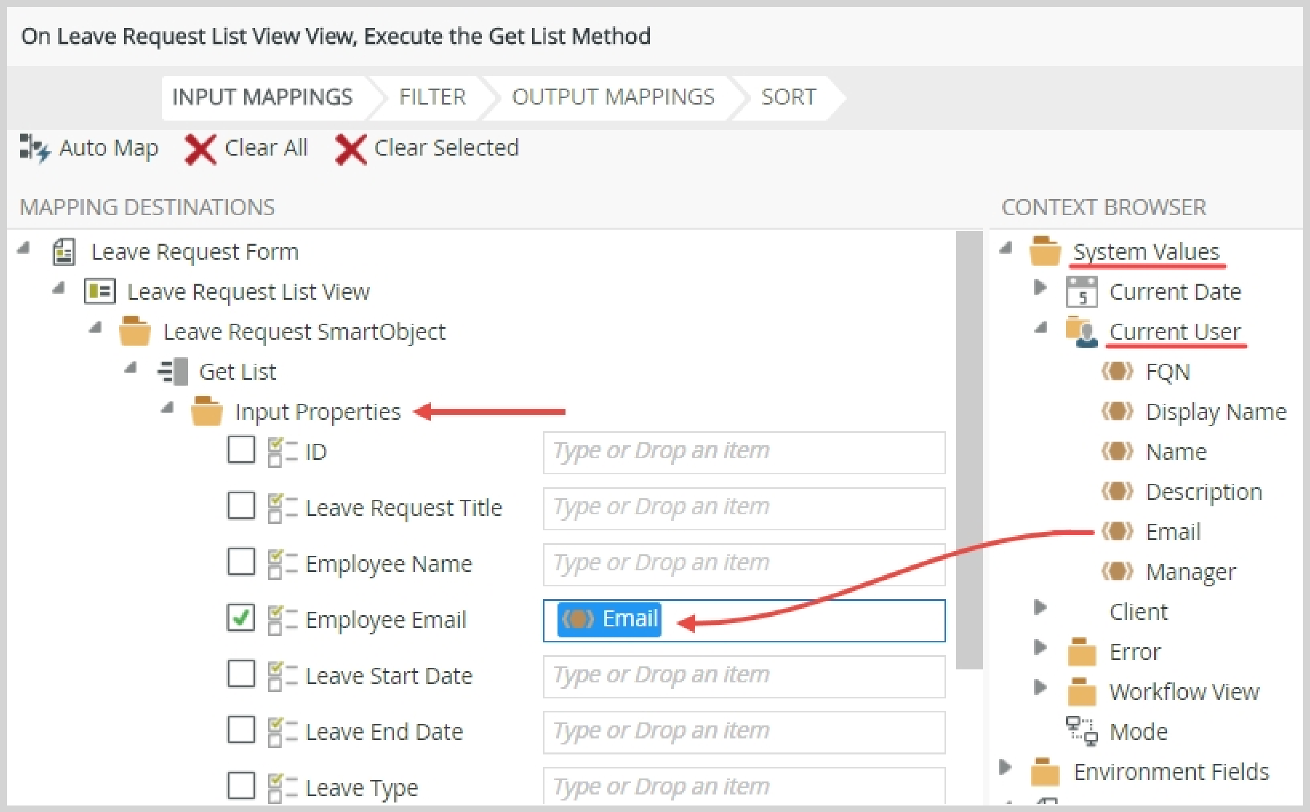
then on Leave Request Item View View, execute the Create method

then start the Leave Request Workflow Workflow (configure)

then on Leave Request Item View View, execute the Clear method (configure)

then on Leave Request List View View, execute the Get List method (configure)

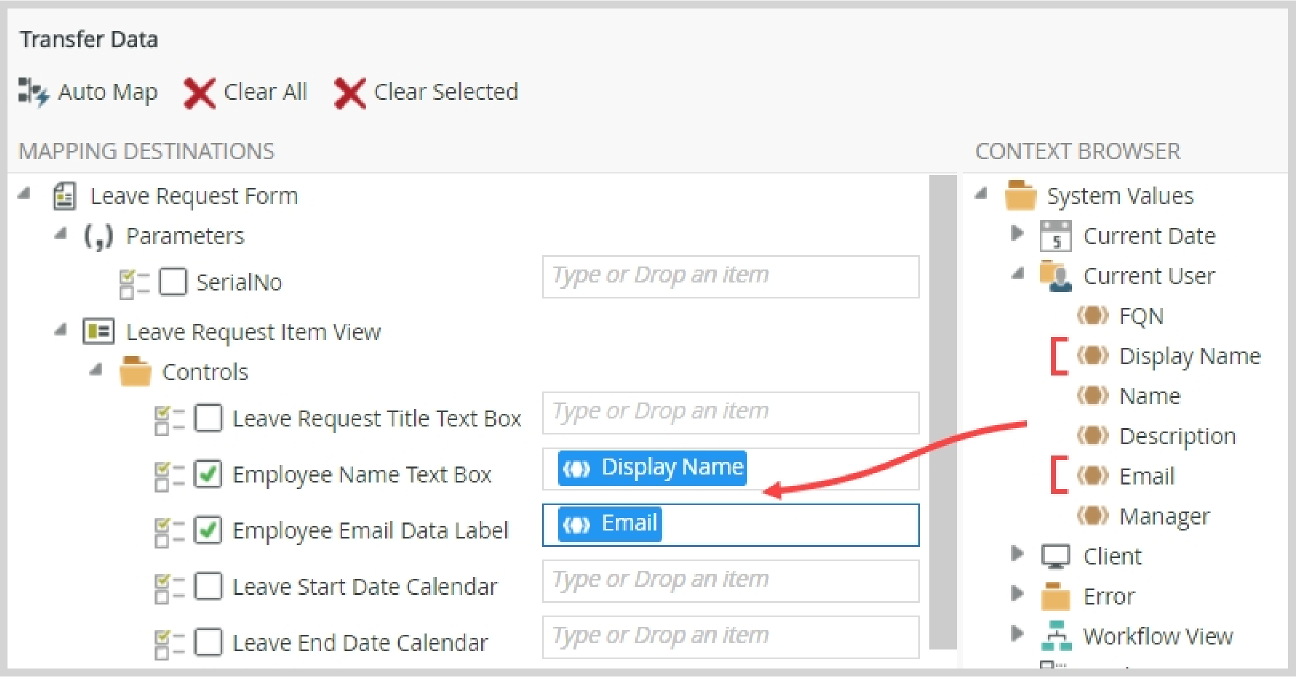
b. Now you will map the current user's email as the input property. This step tells K2 to search the Leave Request SmartObject records for those that have the same email value as the current user. In the CONTEXT BROWSER, expand **System Values**, then **Current User** and drag the **Email** property into the **Employee Email** input property. Click **FINISH**.



On Leave Request List View View, Execute the Get List Method

☐ ID	Type or Drop an item
☐ Leave Request Title	Type or Drop an item
☐ Employee Name	Type or Drop an item
☑ Employee Email	Email
☐ Leave Start Date	Type or Drop an item
☐ Leave End Date	Type or Drop an item
☐ Leave Type	Type or Drop an item

3. Because you just cleared all form fields after you submitted the form, the name and email fields will no longer display the current user. (In the basic tutorial, you configured the name and email fields to pre-populate with the current user's details.) In this step, you will pre-populate the name and email fields once again with system values. Add a **transfer data** action and map the **System Values** > **Current User** > **Display Name** and **Email** to their respective item view controls. **Finish** the form.
   Last, **check in** the views and forms in your application.



Transfer Data

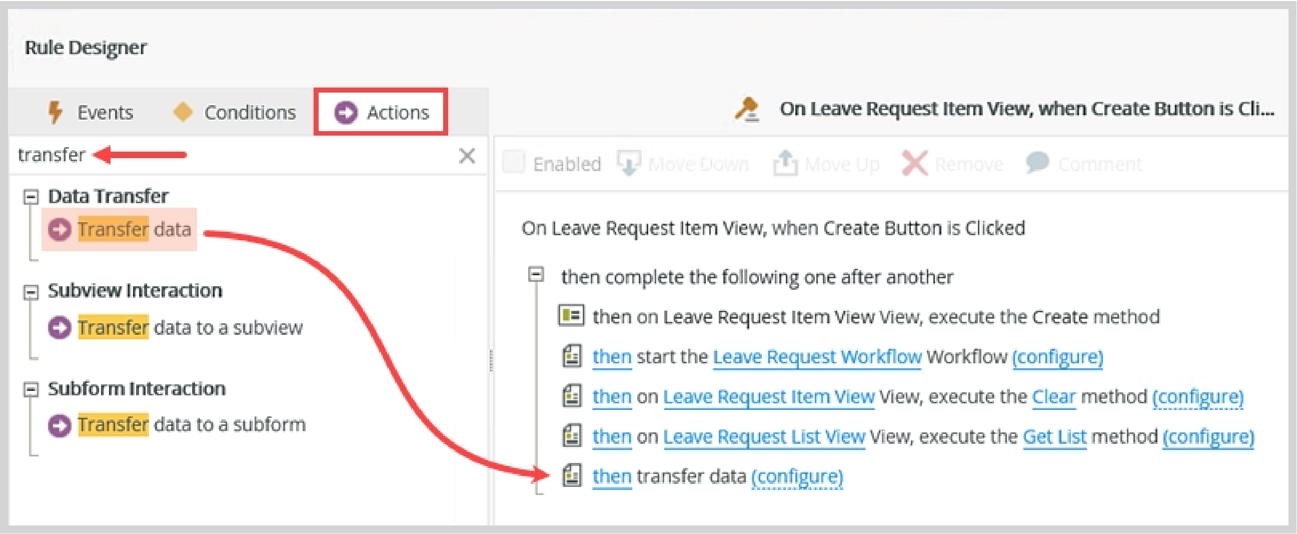
MAPPING DESTINATIONS

- Leave Request Form
  - Parameters
    - □ SerialNo — Type or Drop an item
  - Leave Request Item View
    - Controls
      - □ Leave Request Title Text Box — Type or Drop an item
      - ☑ Employee Name Text Box — Display Name
      - ☑ Employee Email Data Label — Email
      - □ Leave Start Date Calendar — Type or Drop an item
      - □ Leave End Date Calendar — Type or Drop an item

CONTEXT BROWSER

- System Values
  - Current Date
  - Current User
    - FQN
    - Display Name
    - Name
    - Description
    - Email
    - Manager
  - Client
  - Error
  - Workflow View

   a. There is one more step before you can save and exit the form. In the basic version of this tutorial, you configured an action that populates the current user's name and email fields when the form loads. The clear method configured in the previous steps clears the name and email values. You will add a transfer data action to repopulate the name and email fields when the form reloads.
      With the **Actions** tab still active, search for
      transfer
      then click **Transfer data** to add it to the rule definition pane. Click the **(configure)** link.



Rule Designer

transfer

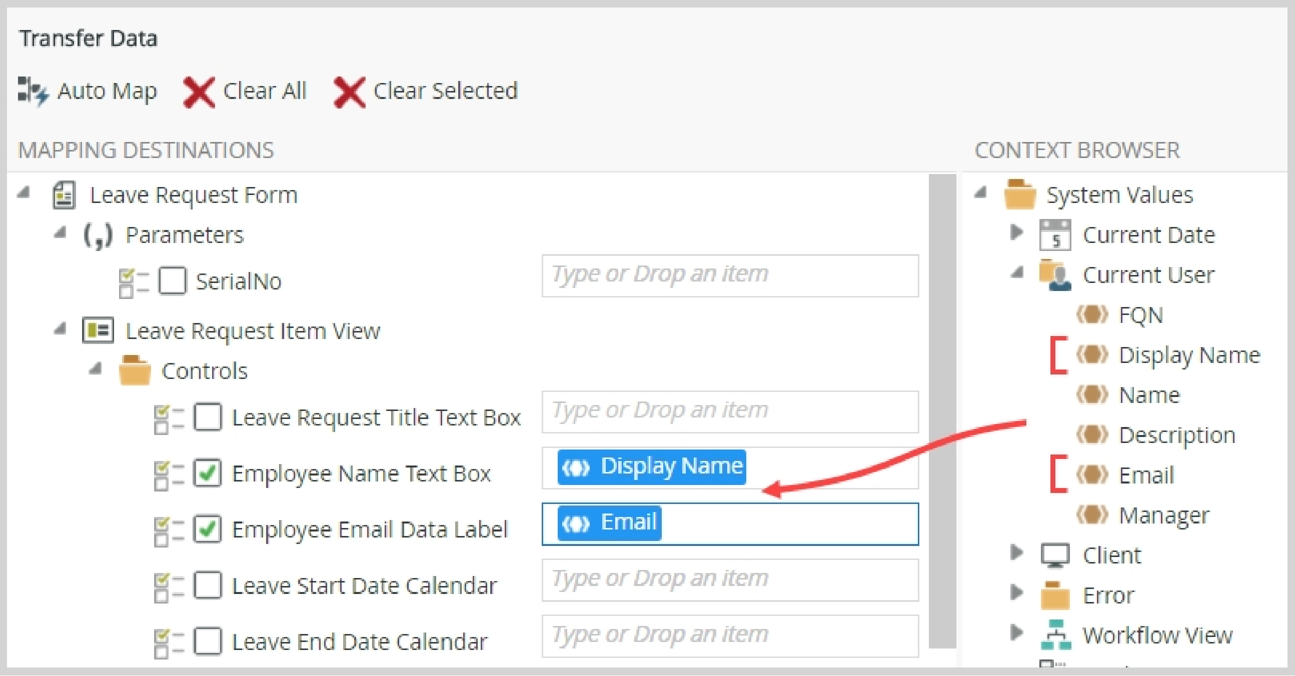
- Data Transfer
  - Transfer data
- Subview Interaction
  - Transfer data to a subview
- Subform Interaction
  - Transfer data to a subform

On Leave Request Item View, when Create Button is Cli...

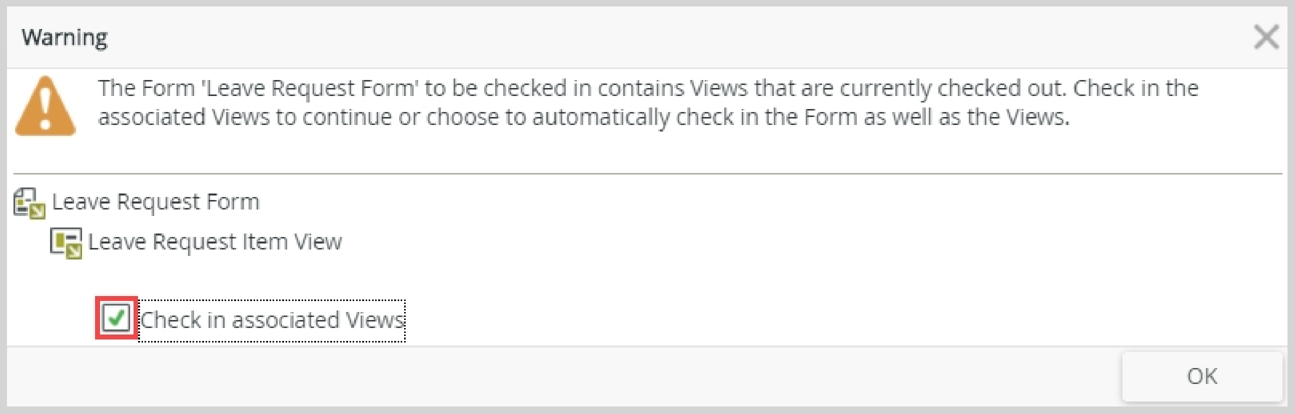
On Leave Request Item View, when Create Button is Clicked

then complete the following one after another
- then on Leave Request Item View View, execute the Create method
- then start the Leave Request Workflow Workflow (configure)
- then on Leave Request Item View View, execute the Clear method (configure)
- then on Leave Request List View View, execute the Get List method (configure)
- then transfer data (configure)

b. In the CONTEXT BROWSER, expand the **System Values** node, then **Current User**. Drag the **Display Name** into the **Employee Name Text Box** and the **Email** into the **Employee Email Data Label**. Click **OK**, then click **OK** once again. Click **FINISH** to save and exit the form. (The form finish button is near the upper right corner of the screen.)



c. Right-click the **Leave Request Form** and select **Check In**. If you see a warning about associated views, CHECK the box to include the views, then click **OK**. Finishing the form "publishes" it to the K2 server. The changes you made in the last few steps are now ready for use!


Warning

The Form 'Leave Request Form' to be checked in contains Views that are currently checked out. Check in the associated Views to continue or choose to automatically check in the Form as well as the Views.

Review

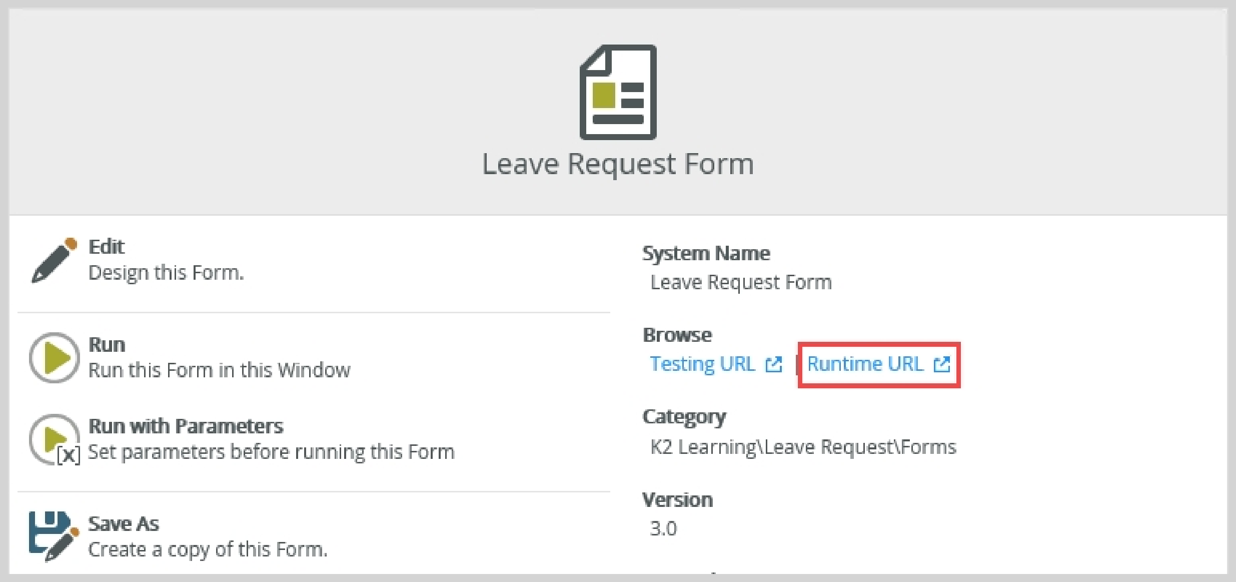
In this step, you edited the rule that fires when you click the create button. You added a clear method that will remove the form field values after you click the button. This lets you know that your form submitted successfully and also provides a clean form if you want to submit another leave request. Because the clear method removed all field values, you added a transfer data action to replace the user's name and email. Finally, you added a get list method to update the Leave Request List View with the new leave request record.

Next Step: 12. Test the Leave Request (Extended Version) Application

12. Test the Leave Request (Extended Version) Application

Now we are ready to test the Leave Request (Extended Version) Application by submitting two new leave requests. After you submit the requests, you will access the approving manager's email. From the task notification email the manager's receives, you will open the request form, then select "Rework" for your decision. From the requester's email, you will open the rework task notification email, then resubmit the request. The last steps will be to either approve or deny the request, once again as the approving manager, then confirm the decision notification email reflects the correct decision.

1. Using the Runtime URL, submit two Leave Request Forms. Make the **Leave Start Date** for one of the requests today's date. This will fire off the reminder since you have it configured to start within two days of the Leave Start Date. You should see two email notifications, one being the task notification email and the second being the reminder email.
   a. Return to K2 Designer. Click to highlight the **Leave Request Form**. From the properties pane, click the **Runtime URL**. This will launch the form in a web browser. Remember too, this is the external link to the Leave Request Form. Use this link on web pages or emails to provide access to this form.



Leave Request Form

Edit
Design this Form.

Run
Run this Form in this Window

Run with Parameters
Set parameters before running this Form

Save As
Create a copy of this Form.

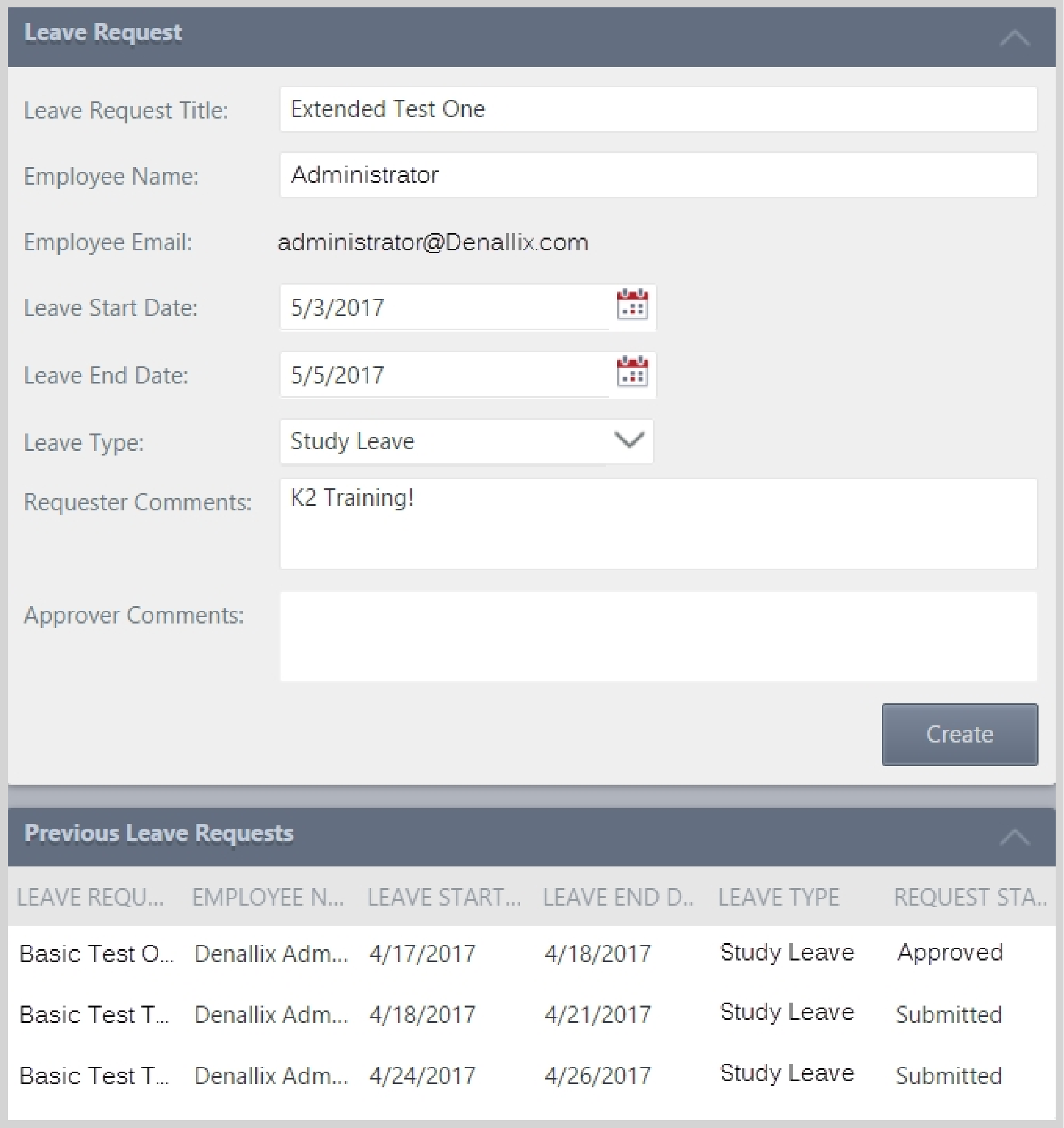
System Name
Leave Request Form

Browse
Testing URL | Runtime URL

Category
K2 Learning\Leave Request\Forms

Version
3.0

   b. You should see some leave requests in the **Previous Leave Requests** list view from testing the Leave Request Basic application. Complete the form fields, making sure that the **Leave Start Date** is today. This fires off the reminder email, which you will see later. Confirm that you cannot edit the **Approver Comments** text area. This control is read-only for the requester. Click **Create**.



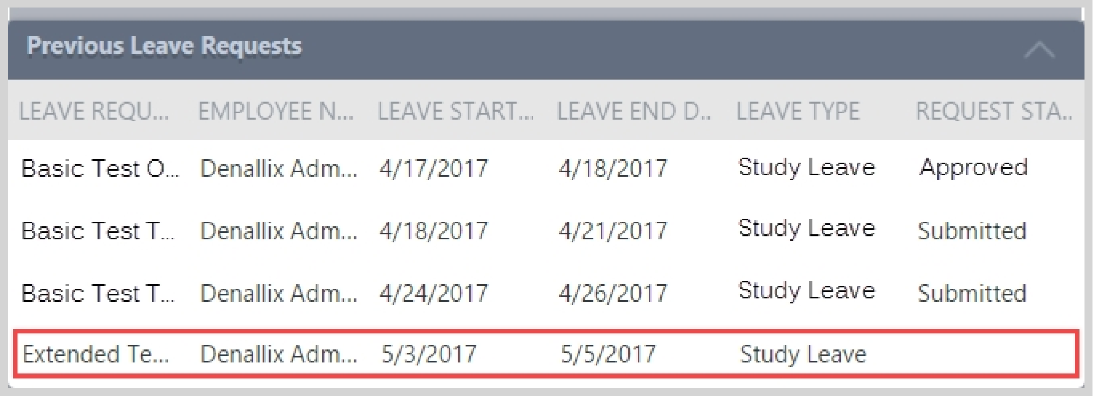
Leave Request

Leave Request Title:	Extended Test One
Employee Name:	Administrator
Employee Email:	administrator@Denallix.com
Leave Start Date:	5/3/2017
Leave End Date:	5/5/2017
Leave Type:	Study Leave
Requester Comments:	K2 Training!
Approver Comments:	

Previous Leave Requests

LEAVE REQU...	EMPLOYEE N...	LEAVE START...	LEAVE END D...	LEAVE TYPE	REQUEST STA...
Basic Test O...	Denallix Adm...	4/17/2017	4/18/2017	Study Leave	Approved
Basic Test T...	Denallix Adm...	4/18/2017	4/21/2017	Study Leave	Submitted
Basic Test T...	Denallix Adm...	4/24/2017	4/26/2017	Study Leave	Submitted

When the form submits, the form fields should clear out and the Employee Name and Employee Email values should repopulate. The leave request you submitted should appear in the **Previous Leave Requests** list view.

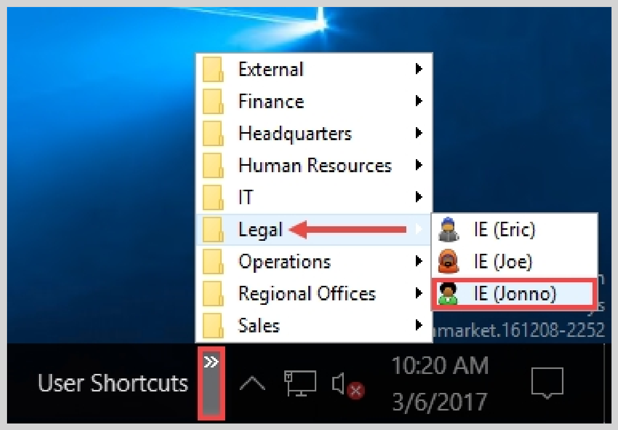


Previous Leave Requests

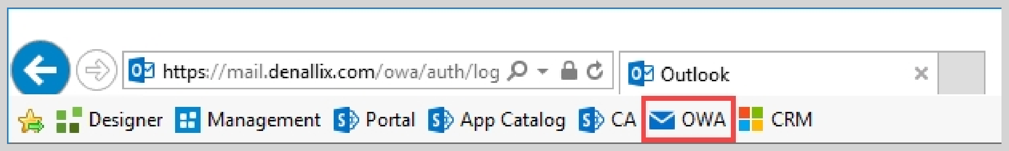
LEAVE REQU...	EMPLOYEE N...	LEAVE START...	LEAVE END D...	LEAVE TYPE	REQUEST STA...
Basic Test O...	Denallix Adm...	4/17/2017	4/18/2017	Study Leave	Approved
Basic Test T...	Denallix Adm...	4/18/2017	4/21/2017	Study Leave	Submitted
Basic Test T...	Denallix Adm...	4/24/2017	4/26/2017	Study Leave	Submitted
Extended Te...	Denallix Adm...	5/3/2017	5/5/2017	Study Leave	

c. **Submit** one or more leave requests. Use any start date you choose. **Close** the browser tab.

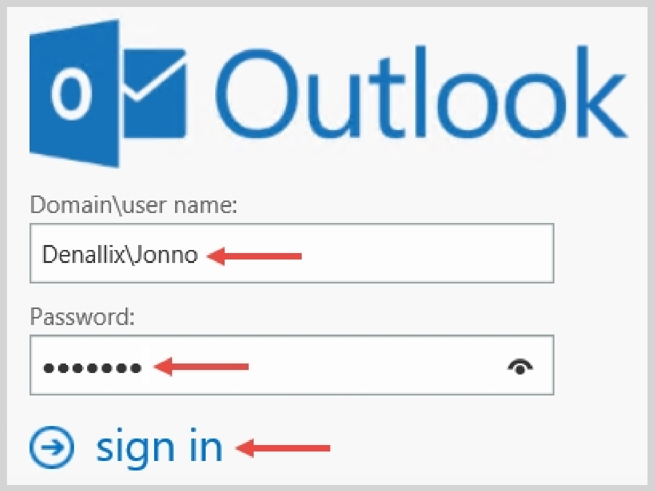
2. Access the approving manager's email and using the **Worklist Item** link, open the approval form. Select **Rework** for the action and submit the form. For this test, you will send the request back to the leave requester, who must resubmit or cancel the request.
   Next, you will access the manager's email to view the task notification emails. If you are working on a K2-provided VM, you are likely logged in as Denallix Administrator. Administrator's manager is Jonno, so you will open Jonno's Outlook. If you are working in your own environment, access the email account for the approving manager. The screen shots and instructions below reflect a K2 VM environment.
   a. Click the **User Shortcuts** arrows found in the lower-right corner of the screen. Expand the **Legal** folder. Click **IE (Jonno)**.



   b. Click the **OWA** (Outlook Web Access) link in the Favorites bar.



   c. When OWA launches, enter Jonno's login credentials.
   Domain\user name: *Denallix\Jonno*
   Password: *K2pass!*
   then click **sign in**.

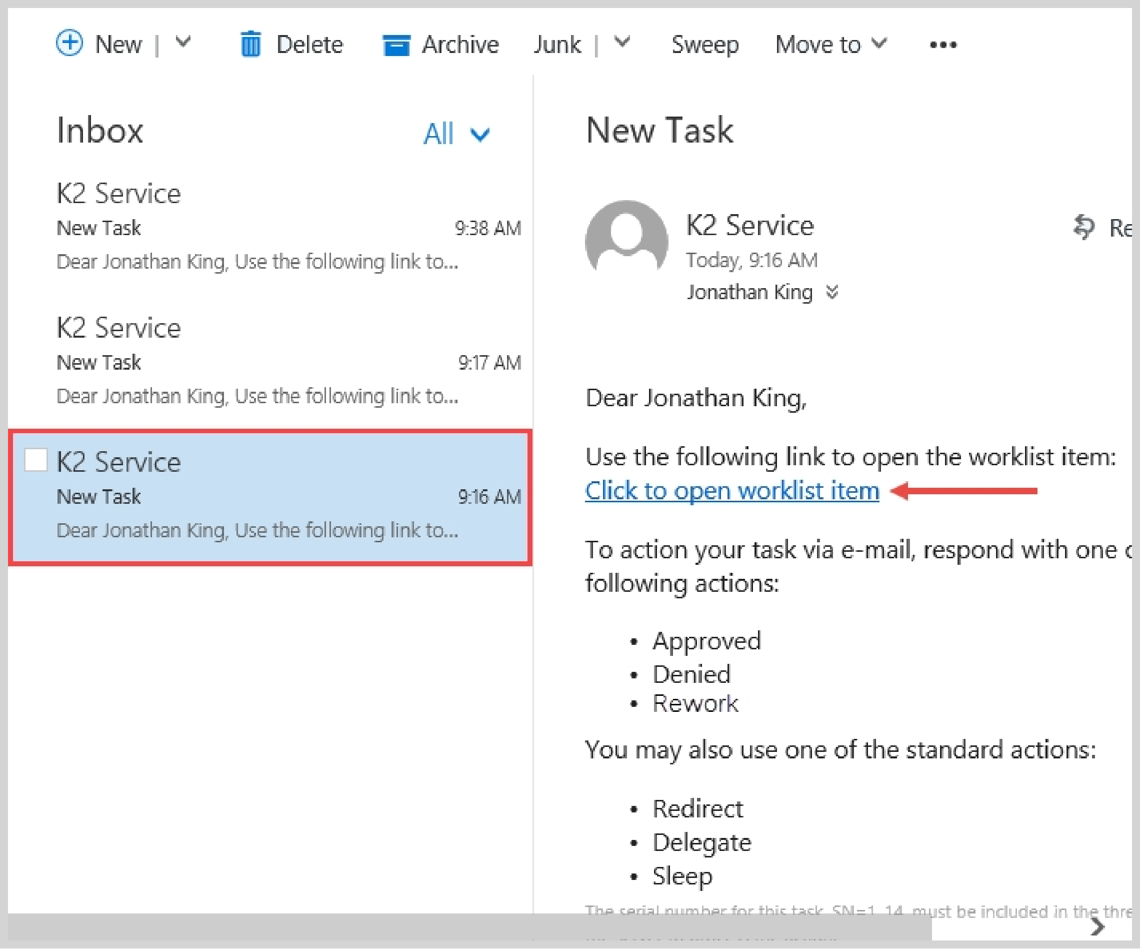


Domain\user name: Denallix\Jonno

Password: •••••••

sign in

d. You should see task notification emails for each of the leave requests you submitted. You should also see the escalation email. **Open** one of the emails, then click the **worklist item** link to open the manager's approval form.



K2 Service
New Task 9:38 AM
Dear Jonathan King, Use the following link to...

K2 Service
New Task 9:17 AM
Dear Jonathan King, Use the following link to...

K2 Service
New Task 9:16 AM
Dear Jonathan King, Use the following link to...

New Task

K2 Service
Today, 9:16 AM
Jonathan King

Dear Jonathan King,

Use the following link to open the worklist item:
Click to open worklist item

To action your task via e-mail, respond with one o
following actions:

- Approved
- Denied
- Rework

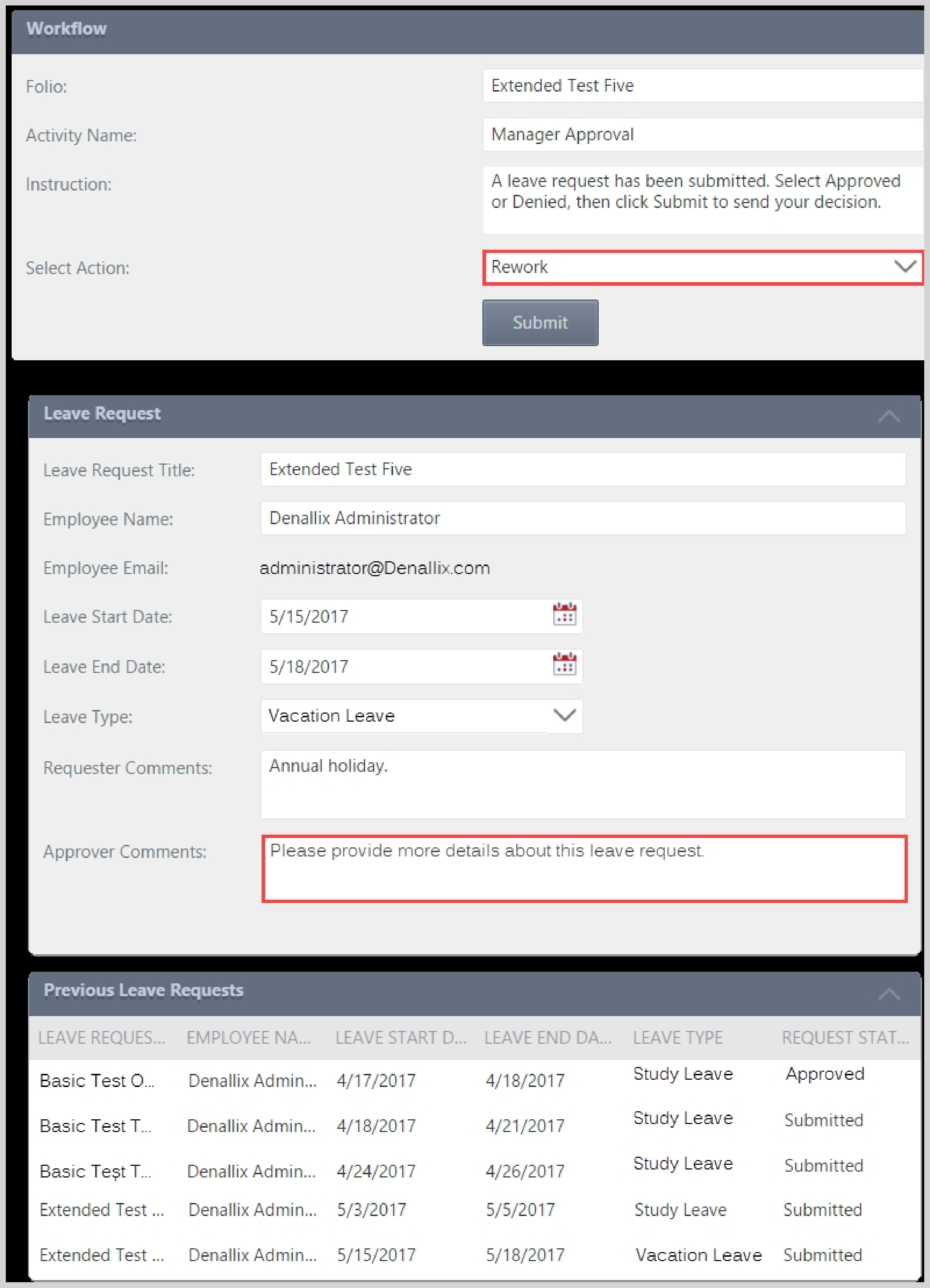
You may also use one of the standard actions:

- Redirect
- Delegate
- Sleep

The serial number for this task, SN=1_14, must be included in the thre

e. Review the request details. Notice that K2 has added the **Workflow** view to the top of the form. This view contains the actions available to the approving manager. From the **Select Action** list, choose **Rework**. Before you submit your decision, enter a few comments into the **Approver Comments** text area. This will let the requester know why you are sending the request back for rework. Click **Submit**.

Close the browser tab.



Workflow

Folio:	Extended Test Five
Activity Name:	Manager Approval
Instruction:	A leave request has been submitted. Select Approved or Denied, then click Submit to send your decision.
Select Action:	Rework

Leave Request

Leave Request Title:	Extended Test Five
Employee Name:	Denallix Administrator
Employee Email:	administrator@Denallix.com
Leave Start Date:	5/15/2017
Leave End Date:	5/18/2017
Leave Type:	Vacation Leave
Requester Comments:	Annual holiday.
Approver Comments:	Please provide more details about this leave request.

Previous Leave Requests

LEAVE REQUES...	EMPLOYEE NA...	LEAVE START D...	LEAVE END DA...	LEAVE TYPE	REQUEST STAT...
Basic Test O...	Denallix Admin...	4/17/2017	4/18/2017	Study Leave	Approved
Basic Test T...	Denallix Admin...	4/18/2017	4/21/2017	Study Leave	Submitted
Basic Test T...	Denallix Admin...	4/24/2017	4/26/2017	Study Leave	Submitted
Extended Test ...	Denallix Admin...	5/3/2017	5/5/2017	Study Leave	Submitted
Extended Test ...	Denallix Admin...	5/15/2017	5/18/2017	Vacation Leave	Submitted

3. Access the requester's email and using the **Worklist Item** link, open the form. Edit the form entry in any manner you like, then resubmit it.
   a. Now you will switch back and view the email for the requester to rework their request. Open **Outlook** for the currently loggged-in user. Confirm there is a Rework task notification email for the user that submitted the leave request. Click the **Click to open worklist item** link so that you can open and edit your original request. (Remember if you are working in your own environment, you most likely will need to open your own Outlook.)
   b. Make a change of some kind to your form entries. Select **Resubmit** from the Action options and click **Submit**. You should see a confirmation dialog, click **OK**. Keep Outlook open for now.
4. Access the approving manager's email and using the **Worklist Item** link, open the approval form. Add some comments in the Approver Comments text area box, then select either Approved or Denied and submit your decision. From the requester's email, confirm the email notification you receive contains the correct decision.
   a. Switch back to the managers Outlook (in a K2-provided environment this should be Jonno). Once again, you should see a new task notification email. This time, you will respond using SmartActions. **Reply** to the email with either **Approved** or **Denied** as the message body. Send the email.
   b. Switch back to **Administrator's Outlook**. Confirm the final email arrived and it reflects the correct decision made by the manager.

Review

In this step, you tested your extended version of the Leave Request application. You learned how you can add properties to existing SmartObjects, then use those properties as view controls. You learned how to use a data source for drop-down list values, and finally, you learned how to implement a rework loop to your workflow. In the next optional step, you learn how you can clean your K2 environment by deleting unused artifacts.

Next Step: 13. (Optional) Clean Your K2 Environment by Deleting Unused Artifacts